AN ALLOCATION BASED MODELING AND SOLUTION FRAMEWORK FOR LOCATION PROBLEMS WITH DENSE DEMAND

Ekrem Alper MURAT

Desautels Faculty of Management

McGill University, Montreal

July 2005

A thesis submitted to the
Faculty of Graduate Studies and Research
In partial fulfillment of the requirements of the degree of
Doctor of Philosophy

Your file Votre référence
ISBN: 978-0-494-27822-2
Our file Notre référence
ISBN: 978-0-494-27822-2

Abstract

In this thesis we present a unified framework for planar location-allocation problems with dense demand. Emergence of such information technologies as Geographical Information Systems (GIS) has enabled access to detailed demand information. This proliferation of demand data brings about serious computational challenges for traditional approaches which are based on discrete demand representation. Furthermore, traditional approaches model the problem in location variable space and decide on the allocation decisions optimally given the locations. This is equivalent to prioritizing location decisions. However, when allocation decisions are more decisive or choice of exact locations is a later stage decision, then we need to prioritize allocation decisions. Motivated by these trends and challenges, we herein adopt a modeling and solution approach in the allocation variable space.

Our approach has two fundamental characteristics: Demand representation in the form of continuous density functions and allocation decisions in the form of service regions. Accordingly, our framework is based on continuous optimization models and solution methods. On a plane, service regions (allocation decisions) assume different shapes depending on the metric chosen. Hence, this thesis presents separate approaches for two-dimensional Euclidean-metric and Manhattan-metric based distance measures. Further, we can classify the solution approaches of this thesis as constructive and improvement-based procedures. We show that constructive solution approach, namely the shooting algorithm, is an efficient procedure for solving both the single dimensional n-facility and planar 2-facility problems. While constructive solution approach is analogous for both metric cases, improvement approach differs due to the shapes of the service regions. In the Euclidean-metric case, a pair of service regions is separated by a straight line, however, in the Manhattan metric, separation takes place in the shape of three (at most) line segments. For planar 2-facility Euclidean-metric problems, we show that shape preserving transformations (rotation and translation) of a line allows us to design improvement-based solution approaches. Furthermore, we extend this shape preserving transformation concept to n-facility case via vertex-iteration based improvement approach and design first-order and second-order solution methods. In the case of planar 2-facility Manhattan-metric problems, we adopt translation as the shape-preserving transformation for each line segment and develop an improvement-based solution approach. For n-facility case, we provide a hybrid algorithm. Lastly, we provide results of a computational study and complexity results of our vertex-based algorithm.

Résumé

Dans cette dissertation, nous présentons une approche unifiée à la résolution du problème d'emplacement et d'attribution (location-allocation problem) avec forte demande. L'émergence de technologies d'information telles les systèmes d'information géographique (SIG) a rendu possible l'accès à une information détaillée sur la demande. Cette prolifération de l'information sur la demande entraîne de sérieux défis de calcul pour les approches traditionnelles, lesquelles sont basées sur une représentation discrète de la demande. À la lumière de ces nouvelles tendances et nouveaux défis, nous adoptons dans le présent ouvrage une approche de modélisation et de solution dans l'espace de variable d'attribution.

Notre approche comporte deux caractéristiques fondamentales: la représentation de la demande sous forme d'une fonction de densité continue et des décisions d'attribution sous forme de régions de services. Sur un plan, des régions de services (décisions d'allocation) prennent des formes différentes selon les métriques sélectionnées. Les approches de solutions proposées peuvent également être classifiées en termes de procédures dites « constructives » ou basées sur l'amélioration. Alors que l'approche à solution constructive est similaire pour les deux cas de métriques, l'approche basée sur l'amélioration diffère dans les deux cas en raison des formes des régions de services. Dans le cas de la métrique Euclidienne, une paire de régions de services est séparée par une ligne droite, alors que dans le cas de la métrique de Manhattan, la séparation s'effectue sous la forme de 3 segments linéaires, au plus. Pour les problèmes de

métrique Euclidienne à 2 installations, nous démontrons que des transformations préservant la forme d'une ligne (rotation-translation) nous permettent de formuler des approches de solutions basées sur l'amélioration. De plus, nous développons ce concept de transformation préservant la forme dans le cas à n installations via une approche d'amélioration basée sur des itérations vertex et formulons des méthodes de solution d'ordre de premier et second ordre. Dans le cas du problème à 2 installations sur un plan avec métrique de Manhattan, nous adoptons la translation comme transformation préservant la forme pour chaque segment linéaire and développons une approche de solution basée sur l'amélioration. Cependant, nous démontrons que cette adaptation de l'approche à itération-vertex pour les problèmes de métrique de Manhattan à n installations se révèle difficile à mettre en œuvre. Ainsi, nous développons un algorithme hybride.

Contents

Acknowledgments

This thesis would not be possible without my advisor Vedat Verter and his patience, guidance on putting things into perspective and teaching me the importance of conveying the most theoretical concepts in easily understandable forms. Throughout this journey not only he financially supported my studies, but also showed extreme attention to my development as a researcher. For all his passion and empathy for his students, I feel an unlimited gratitude for his advising my study.

I also wish to thank to my committee members. Gilbert Laporte has stood by me throughout the process and provided invaluable advice and directions which constantly shaped and matured my dissertation. Without his advices, I believe this dissertation would not be as mature as its current state. Tamer Boyaci has provided me the support and guidance whenever I needed. Despite the distance constraint, Alain Martel has always been supportive and got my interest into the stochastic programming which has become part of my future research agenda.

Friends, colleagues and comrades in the department have always been great support. Among all, Catherine has put up with my messy desks and agitated approach to research without a single word of complaint. Without her supportive attitude, the office would not be the most productive research venue. Marilyn Gombe, our area secretary has been a great facilitator for the bureaucratic processes and a great person to have casual conversations.

Of course, this pursuit would not be complete or smooth without the understanding and support of my family. For that, I devote this dissertation to my family.

LIST OF FIGURES

LIST OF TABLES

Chapter 1

1.1 Introduction

Location-allocation problems arise in many different decision making environments ranging from supply chain network design in logistics to clustering methods in data mining. In the context of logistics, a decision maker of a location-allocation problem is concerned with the optimization of system-wide costs of locating a number of facilities (servers) to supply a given set of customers, whose location and demand information are known. This problem involves two types of decisions: Location of the facilities and allocation of customers to the facilities. While the allocation decisions can be changed over time, location decisions often require huge investments, thus considered as part of strategic planning. Location-allocation problem has two common forms: Discrete and planar problems. Discrete location-allocation problem, known as p-median problem, has a set of predefined facility locations. Planar problems, however, allow location of facilities on any point in the plane. In this dissertation, we focus on the planar location-allocation problems.

Traditionally, planar location-allocation problems are modeled with discrete demand representation and solved in the location variable space. In this solution approach, allocation decisions are assumed to be optimally made given the locations. One advantage of this approach is the ease of optimizing allocation decisions for a given set of facility locations by solving the transportation problem.

In this thesis, we develop an alternative framework for location-allocation problems by changing two aspects of the traditional approach: Discrete demand representation and solution in the location space. In our approach, we represent demand as continuous density functions and solve the problem in the allocation space. In comparison with traditional solution methods, our solution approach assumes locations are optimally made given the allocation decisions.

1.2 Motivation

Our motivation in this thesis is two fold. First, demand information has become more abundant with the emergence of new technologies. Over the last decade, successful implementation of such information technologies as point of sales data and Geographical Information Systems (GIS) has enabled access to demand information at a detail level which was not available before. In addition, the scope of the location-allocation problems in the logistics environments has widened due to such economic trends as globalization. Therefore, these developments have motivated us to represent demand information in the form of continuous functions.

Secondly, solving location-allocation problems in the location variable space has some challenges. One challenge is the need for determining location of the facilities a priori. This task involves search, selection and feasibility analysis of candidate sites to be used in the problem. Another challenge is related to the solution procedures. The objective function of the multi-facility location problem with Euclidean-metric is known to be non-differentiable when the facility locations overlap with each other or with demand points. This challenge has attracted the interest of many researchers who attempt various approximation and heuristic methods to overcome it. Hence, these challenges have motivated us to solve the planar location-allocation problem in the allocation variable space.

1.3 Organization of Remainder of the Thesis

In order to solve the planar location-allocation problem in allocation variable space with continuous demand density functions, we developed various models and solution methodologies. These models and methods are part of an unified framework, but differ from each other for different metrics and number of facilities. Table 1.1 illustrates various model types and solution methods by their corresponding chapter.

Type	Algorithm	Single Dimensional Location-Allocation Problem	Planar Location-Allocation problem			
			2-Facility and Euclidean	N-Facility and Euclidean	2-Facility and Manhattan	N-Facility and Manhattan
Constructive Solution	Shooting Algorithm	Chapter 3	Chapter 4		Chapter 7	
Improvement Based Solution	Sequential Location-Allocation (SLA) with Continuous Demand		Chapter 4*	Chapter 4*	Chapter 4*	Chapter 4*
	Steepest-Descent Method for Allocation Decisions	Chapter 3	Chapter 4		Chapter 7	
	Modified Newton's Method for Allocation Decisions		Chapter 4			
	Steepest-Descent Method based on Vertex-Iterations			Chapter 5		
	Conjugate-Gradient Method based on Vertex-Iterations			Chapter 5		
	Hybrid Improvement Algorithm (Steepest Descent+SLA)	Chapter 7				Chapter 7

Table 1.1: Overview of the problem types and algorithms

In Chapter 3, we focus on the single-dimensional problem and develop an alternative model for the location-allocation problem, i.e., dynamic programming model based on the allocation decisions. To solve this model, we propose two solution methods: Constructive shooting algorithm and improvement based steepest-descent method. Constructive algorithm solves a relaxation of the problem by relaxing the market boundary constraints. In contrast, steepest-descent method, starting with an initial solution, improves the current solution based on the quality of the allocation decisions. This chapter also provides several analytical results for the single-dimensional problem with a linear demand density.

In Chapter 4, we present two solution approaches for the 2-facility planar problem with Euclidean-metric. These methods are the constructive and improvement-based solution approaches, which constitute an extension of those in the single-dimensional setting (Chapter 3). In both of these extensions, we utilize the special form (i.e., straight line) of the allocation decisions for Euclidean-metric problems. Specifically, planar 2-facility constructive solution approach is based on the following property of a straight line: Two points are sufficient to define a line. For the improvement-based steepest descent approach, we couple the line form with shape preserving transformations (rotation and translation) as a way to iterate the allocation decisions. Lastly, we provide a second-order improvement method, i.e. modified Newton's method, to alleviate the linear convergence rate of steepest-descent method. The appendix to this chapter includes a special case of the well-known sequential location-allocation(SLA) method, where the demand is a continuous density function.

In Chapter 5, we first illustrate the challenges associated with extension from 2-facility case (Chapter 4) to n-facility case. The two methods developed in previous chapters bring about different challenges. Extension of the constructive solution approach is impracticable due to the incomplete information of the solution. On the other hand, the main challenge for the improvement-based approaches is that we can no longer iterate from one allocation solution to another by a simple line movement. However in this chapter, we demonstrate that we could iterate the vertices as oppose to the allocation lines. Hence, we are still moving from one allocation solution to another, but this time with the use of vertices. Note that, this transition, from the line-representation to vertex-representation of the allocation decisions, is possible only when the vertices are connected to each other in the form of straight lines (special case for Euclidean-metric). We provide two improvement algorithms for the vertex-iterations: steepest-descent for vertex iterations (first-order method) and conjugate-gradient for vertex iterations (second-order method).

Chapter 6 presents results of a computational study for the planar n-facility Euclidean-metric problem. This study includes various demand density functions and number of facilities. The solution method employed in solving these problems is the vertex-iteration based steepest-descent approach from Chapter 5. Based on the results of this computational study, we provide insights on how demand density (shape and variation), number of facilities and cost parameters affect the objective function as well as the solutions. We also report on the computation complexity (both theoretical and runtime performance) of this method.

Chapter 7 exclusively considers the problems based on the Manhattan-metric. In this chapter, building on the results from previous chapters, we develop models and methods for various problem types, i.e. single-dimensional problem, planar 2-facility problem and planar n-facility problem. Organizationally, this chapter resembles the development from chapter 3 until chapter 5. First, we develop a special improvement-based solution method on a line. This method is the steepest-descent method, which is based on the iteration of optimal locations as surrogate iterates of the allocation decisions. Then, we extend this approach to planar n-facility problems by combining the sequential location-allocation approach and steepest-descent method. We refer to this composite algorithm as hybrid improvement algorithm. For planar 2-facility problems, we extend the results of Chapter 4 by accounting for the special structure of the Manhattan-metric's allocation line. In contrast to the straight line in Euclidean-metric, Manhattan-metric allocation line for 2-facility is composed of at most three line segments. After adjusting for this difference, we extend both the constructive and the improvement-based solution approached of Chapter 4 for Manhattan-metric.

Since the location-allocation problem is a non-convex optimization problem and has many local optimal solutions (with the exception of such special cases as on a line with linear demand density), term "optimal solution" in this thesis refers to local optimal solutions unless otherwise stated.

Chapter 2

Literature Review

2.1 Introduction

In this review, we present some of the earlier work on planar minisum location and location-allocation problems. Since the literature on planar models is enormous, we do not claim to be comprehensive in our review. Our goal is to provide references on the studies which relate parts of the planar minisum problems addressed in this thesis. Other reviews of interest are Francis et al (1983), Brandeau and Chiu (1989), Avella et al. (1998), Hamacher and Nickel (1998), Scaparra et al. 2001, Wesolowsky (1993) and Hale and Moberg (2003) for general location science, Okabe and Suzuki (1997) for Voronoi diagram approaches to various location problems, Agarwal and Sharir (1998) for geometric optimization, and Langevin et al. (1996) for continuous approximation of distribution problems. There are also several books and chapters covering the problem in this chapter. Some of these books are Drezner (1995), Francis and White (1998).

Before presenting the review of different problems and applications, it is important to mention the aggregation technique which aims to remedy the complexity of the location-allocation problems by reducing the size of the demand data set through aggregations. The literature on this topic is quite extensive, and therefore, we provide pointers for the references which analyze the techniques in more detail. Francis et al. (2005) provide the most recent review of the literature on aggregation. An empirical study on the comparison of different error bounds is provided in Norman et al. (1995). Zhao and

Batta (1999) study the error bound for the centroid aggregation effect on the Euclidean distance p-median planar location problem. Erkut and Bozkaya (1999) study aggregation errors for p-median location problem. They introduce several new error sources due to the poor choice made by the analyst. To summarize the findings of these studies, aggregation aims to reduce the problem complexity by clustering groups of demand points. In doing so, obviously, there are some errors brought into the problem and solution. In order to minimize these errors, analyst must try to find the best aggregation which requires another step of optimization dedicated towards reducing the aggregation error. This is called the paradox of aggregation which turns a single optimization problem into two-step optimization problem. The absence of consensus on the universal error measures is another issue, which limits the applicability of this procedure.

The organization of this chapter is as follows. In section 2.2., we review pure facility location problems (i.e., allocation decisions are fixed) for discrete demand and continuous demand cases. In section 2.3., we present the literature on location allocation problems with discrete and continuous demand. Section 2.4. is dedicated to the continuous approximation approach for location-allocation problems. Section 2.5. reviews the literature on the voronoi diagram approach for location-allocation problems.

2.2 Pure Location Problems

This problem deals with the determination of the location of a single facility to minimize the sum of weighted distances to demand points. Single facility location problems are also known as the Fermat-Steiner-Weber problems

(Drezner, 1995). In this section, we will review the related literature based on the two main metrics used in this chapter, namely the Manhattan (or rectilinear metric) and the Euclidean-metric as well as generalized metrics. This section also presents the pure location studies for continuous demand cases.

2.2.1 Discrete Demand and Manhattan-metric

In the transportation literature, Manhattan metric is known to estimate the distance traveled in city blocks better than the Euclidean metric. This measure also preferred in the VLSI circuit design and placement of equipment on the shop floor. The objective function is piecewise-linear and convex. For the Manhattan-metric, Francis (1963) and Francis and White (1998) show that exact solutions are found at median locations, i.e. half of the demand lies on the left(above) and other half on the right(below). Vergin and Rodgers (1967) provide a test for identifying these median locations. For multi-facility case, Cabot et al. (1970) provide a minimum-cost network flow approach whereas Wesolowsky and Love (1971b) provide a linear programming formulation after introducing additional variables to replace the absolute term in the objective. However, these linear programming formulations proved to be large for the algorithms and computers of that time. Accordingly, Pritsker and Ghare (1970, 1971) provide a gradient based approach to overcome this challenge. Pritsker (1973) made a correction in the algorithm and provided upper and lower bounds of the objective function based on the Euclidean distance problem. Rao (1973) showed that Pritsker and Ghare (1970, 1971) may not converge to optimal solution with counter examples. Juel and Love (1976) extend Pritsker and Ghare (1970, 1971) by providing necessary and sufficient conditions and a modification to the earlier method which guarantees convergence. Picard and Ratliff (1978) provide an approach where they solve at most (m-1) minimum-cut flow problems on networks with at most (n+2) vertices. They label their approach as the primal method and compare with dual approach, i.e. minimum cost network flow approach in Cabot et al. (1970). Wesolowsky and Love (1972) provide a reduced gradient solution approach after approximating the objective with hyperboloid functions. Another hyperboloid approximation is proposed by Eyster et al. (1973). Sherali and Shetty (1978) improve the combinatorial step in Juel and Love (1976)'s approach which enumerates the subsets of facilities to be moved. Specifically, they solve a quadratic binary integer problem for the subset identification. Kolen (1981) shows the equivalence of the direct search approach in Prikster and Ghare (1970), Juel and Love (1976), and Sherali and Shetty (1978) to the cut approach in Picard and Ratliff (1978). Cheung (1980) modifies the approach in Picard and Ratliff (1978) by reducing the size of the network problems and tightening the feasible space.

2.2.2 Discrete Demand and Euclidean-metric

When the problem is single-facility location on a line, it is known that the optimal location coincides with one of the demand locations. Therefore, in the planar setting it is assumed that the demand points are non-collinear.

When the demand points are not collinear, the objective of single facility problem with Euclidean metric is known to be strictly convex (Francis and White, 1998). The first method for solving this problem is due to Weiszfeld (1937). Since original work published distantly (in French and in a Japanese journal), it has been rediscovered by Miehle (1958), Kuhn and Kuenne (1962) and Cooper (1963, 1964). Miehle (1958) is the first to extend this approach to multi-facility case. This method generates a sequence of iterates, which are the weighted average of the demand points. In essence, this method is a steepest-descent with inexact step length. In this method, objective is everywhere differentiable except at the existing demand point. Kuhn and Kuenne (1962) provide a modification such that it is everywhere differentiable. Francis and Cabot (1972) provided a dual formulation for the multi-facility Euclidean-metric case. They pointed out that the complementary slackness conditions could provide useful insights.

Kuhn (1973) provides first rigorous analysis of this method. Specifically, it provides conditions for destination optimality (i.e. one of the demand points being the optimal solution) and shows that this method possesses linear convergence as long as the location does not coincide with the existing demand points. It provides a modification in the search direction in the case of such an overlap which guarantees convergence except a denumerable number of starting points. Katz (1974) shows that Weiszfeld's method's convergence rate depends on the optimal location; if the optimal location is not one of the demand points then convergence is linear, or else convergence is either linear, superlinear or quadratic. Ostresh (1978) corrects a flaw in the Kuhn (1973)'s proof and provides a modification step for Weiszfeld's approach to guaranteed convergence when the current iterate coincides with one of the demand points. Ostresh (1978) also provides convergence results for the step length and shows that convergence is guaranteed when the step length is not less than Weiszfeld's step size or greater than double of that. Motivated by the inexactness of the step-size in Weiszfeld's method, Katz and Cooper (1981) propose an optimal gradient method with inexact line search for single facility problem and show it requires less number of iterations than Weiszfeld's method. However, their method requires a line search method for the calculation of step-size. Drezner (1992), in an effort to accelerate the Weiszfeld's method, provides bounds for the variable step length and recommend 1.8 factor as a good adjustment for most problems. Balas and Yu (1982) also modify the Kuhn's method which guarantees convergence for any starting point. Rado (1988) extends Miehle's algorithm and provides a convergence proof based on the theorem first provided in Kuhn (1973) and corrected in Ostresh (1978). Chandrasekaran and Tamir (1989) show that Kuhn (1973)'s convergence proof does not hold unless the convex hull of demand points is in full-dimension of the problem space. Brimberg (1995) provides a formal convergence proof of the Weiszfeld method which states that convergence is guaranteed, except a denumerable number of starting solutions, only if the convex-hull of demand points is in full-dimension.

Ostresh (1977) shows that Miehle's method is indeed a descent method. However, Rosen and Xue (1992) demonstrate that Miehle's algorithm can get stuck at nonoptimal solutions due to the nondifferentiability. In order to come this nondifferentiability, Eyster et al. (1973) extend the Weiszfeld procedure with a Hyperboloid Approximation Procedure (HAP) for multi-facility cases. Charalambous (1985) has accelerates the convergence of HAP. Morris (1981) provides a convergence proof of the HAP procedure for special norms and powers of the distance. Rosen and Xue (1993) show the HAP is a descent procedure and that it always converges from any initial point. Plastria (1992) presents exact optimality conditions when the locations coincide with the demand points. Main contribution of this work is that necessary conditions for overlapping of facility locations and demand points at the optimal solution are generalized.

In addition to the Weiszfeld or HAP, several researchers proposed alternative methods. Francis and Cabot (1972) and Love (1974) provide dual formulations for unconstrained and constrained multi-facility problems, respectively. Plastria (1987) provide a cutting plane method for single-facility problems with mixed norms and convex transportation costs. Calamai and Conn (1980, 1987) and Overton (1983) provide projected Newton methods on the linear manifolds. Their approaches present quadratic convergence; however, with the large problem instances they display poorer performance since the combinatorial nature of explicitly adding and dropping activities in a projection method prove to be computationally expensive. These methods are the first ones to exploit the duality structure of the problem and provide primal and dual solutions at the same time. Vardi and Zhang (2001) provide a monotonically convergent modification to the Weiszfeld method. Xue et al. (1996) points out that the dual formulation suggested by Francis and Cabot (1972) is equivalent to the maximization of a linear function subject to convex quadratic inequality constraints. Accordingly, they suggest solving the dual problem using interior point methods in polynomial order. Another second order approach, Newton bracketing for single-facility location problem, is proposed in Levin and Ben-Israel (2002). Li (1998) propose a Newton-based acceleration for the single-facility case and state that computational results indicate superlinear convergence. Sherali and Al-Loughani (1998) propose reformulations for the multi-facility problem, namely Lagrangian dual and primal space formulation. Their dual formulation is equivalent to Francis and Cabot (1972). Since both formulations are differentiable, they propose using standard continuous optimization methods. Sherali and Al-Loughani (1999) propose a conjugate gradient method for multi-facility problems with two different deflection strategies.

Some of the studies on this problem aim to terminate the solution procedure whenever the solution quality is acceptable. For this, several studies propose bounding schemes. Love and Yeong (1981) and Juel (1984) provide single bounds which can be extended to include multi-facility case. Drezner (1984) proposes a bound for Euclidean-metric case for single facility problems which is later extended to n-facility L2 case by Dowling and Love (1986). Elzinga and Hearn (1983) and Juel (1984) show that Juel (1984) bound is always as good as or better than Love and Yeong (1981) bound. Wendell and Peterson (1984) present a dual based lower bound after constructing a feasible solution to the dual problem a given solution. Love and Dowling (1989a) extend the bound for single-facility generalized metric location problem in Love and Dowling (1989b). They show that it is as good as the Juel (1984) bound and it is equivalent to the bound in Drezner (1984) for Euclidean-metric single-facility problem. Their bound is based on the solution of an alternative rectilinear problem with adjusted weights. Dowling and Love (1987) propose the dual as an alternative lower bound. Uster and Love (2002) provide a rectangular bound for the general metric single and multi-facility cases. However, they do not report on the quality of this bound.

Square of the Euclidean metric (i.e. Squared Euclidean-metric) is mostly used to penalize excessive distances. It possesses the same dimensional separability property as the Manhattan-metric. Eyster and White (1973) cite some special applications for this class of problems. Objective is strictly convex and everywhere differentiable. Eyster and White (1973) analyze this distance measure and present some properties. Since the objective is both separable and differentiable, smooth optimization techniques can be used. Due to the ease of solution, most of the clustering algorithms assume that this metric provides an accurate separation between cluster centers. Francis et al (1998) provides further discussion on this distance measure.

2.2.3 Other Studies with Discrete Demand

In addition, there are other studies which are relevant to the pure location problems. Brimberg and Love (1995a) show that the optimal solution to multi-facility problem is in the convex hull of demand points when cost components are increasing and differentiable function of the norms used to measure the distances. Brimberg and Love (1992) extend the Weiszfeld's procedure to single-facility problems when the Lp metric's "p" is in the closed interval [1,2]. They provide local convergence results and observe that convergence rates are asymptotically linear. Later, Brimberg and Love (1993) provide global convergence proof for this procedure as long as their iterate does not coincide with demand points. They also show that their procedure loses the descent property, thus the convergence, for "p" greater than 2. Drezner and Wesolowsky (1978b) provide a method based on numerical solution of differential equations which is implemented differently for Manhattan and Euclidean metrics. Juel and Love (1983) derive a necessary and sufficient conditions for a demand point to be the optimal location in the case of mixed lp-norms. Ward and Wendell (1980) propose a new norm, called one-infinity, which is a special combination of the Manhattan and Tchebbycheff.

Brimberg and Wesolowsky (2000) consider a different planar location problem, based on rectilinear distances, where the locations of facilities, as well as the demand locations, are considered as areas rather than points. Later, Brimberg and Wesolowsky (2002) extend this formulation with solution procedure to the Euclidean metric. Tuy et al. (1995) consider three variants of the single-facility location problem: with attraction and repulsion (i.e. negative weights), conditional location where there are pre-existing facilities and conditional distances where distance is assumed constant beyond a limit. They reformulate the problem as a "d.c. programming" problem (maximization of differences of convex functions over a convex feasible set). They solve the problem using a triangular branch and bound method.

In addition there are some variants of the location problems which try to relax the restricting assumptions of the planar models location models. Herein, our treatment of these variants serves as providing bibliographic references. Hence, we do not discuss specific contributions of each work listed below nor we claim to be comprehensive in our listing.

Planar models assume that any solution on the plane is a feasible solution. However in practice, this assumption is unrealistic for some special cases. Though it is hard to judge the accuracy, in planar problems, whenever the solution implies an infeasible point, a nearby point is sought as the final destination. If the nearby point is indeed far from the original optimal location, then the inaccuracy worsens. There are a number of factors, such as regions which are inhabitable or affected by zoning regulations, which needs to be accounted for. There are many studies which incorporate these locational constraints as forbidden regions in the planar location formulation. Some example of such studies are Love and Morris (1975) , Hansen et al. (1985), Aykin and Babu (1987), Hamacher and Nickel (1995), Fliege and Nickel (1997), Kafer and Nickel (2001), and Hamacher and Klamroth (2000).

In general planar models assume that the distances are not very long; hence, the planar assumption of the spherical surface of earth is reasonable. However, for very large distances, as in global location decisions, this assumption would give rise to errors. There has been a multitude of studies which consider the geodesic distances on the earth surface. Some of these studies are Drezner and Wesolowsky (1978a), Katz and Cooper (1980), Wesolowsky (1982), Drezner (1985), Aykin and Babu (1987) and Xue (1995).

One major assumption of the planar models is that planar distances are good approximations of the traveled distance on the network. In order to enhance the accuracy of the planar approach, there are several modifications proposed. One of them is to calculate a distance predicting function for the transportation network. The idea is to include coefficient weights for single dimensional distances with a rotation of the reference axis and finding an "ideal" p value for the least inaccuracy. Some of the work in this field is Koshizuka and Kurita (1991), Love and Morris (1979), Love and Walker (1994), Brimberg and Love (1991), Brimberg and Kakhki (2003), Brimberg and Love (1995b), Brimberg et al. (1995), and Uster and Love (2001). In addition, planar location approach assumes that travel costs are proportionate to the distances. There are several studies which relax this assumption by regarding the transportation costs as nonlinear functions of the distance Hansen et al. (1985).

Lastly, planar location problems omit the facility fixed costs in the solution. Brimberg and Salhi (2005) provide a procedure where location dependent fixed costs can be incorporated in the planar problems.

2.2.4 Studies with Continuous Demand

Wesolowsky and Love (1971b) provide a gradient-based search algorithm for Manhattan-metric multi-facility location problem. Love (1972) considers single-facility location problem with rectangular-shaped uniform-density demand areas. He shows that this is a convex optimization problem and solves it using a nonlinear programming technique. Bennett and Mirakhor (1972) consider a similar problem, but recommend using point representation of the area demand which does not have to be rectangular in shape or uniform density. They propose using the centroid as the concentration point. Drezner and Wesolosky (1980) provide an efficient two step solution procedure for Lp metric and general shape demand areas. Odoni and Sadiq (1982) consider the single-facility location problem with Manhattan-metric for uniformly distributed demand over rectangular market region. They show that there are two candidate locations for optimum solution.

Drezner and Drezner (1997) consider competitive single-facility location problem which is based on the gravity model of Huff (1964). They compare the solutions obtained from discrete demand and continuous demand assumption. Their demand is based on the gravity model; hence, it is non-uniform. Their results indicate that the discrete demand case results in more local solution than the continuous case. Accordingly, the objective is much smoother with the continuous demand than discrete. They experiment with 100 starting solutions and observe that discrete demand case does not yield the global solution whereas continuous demand equivalent results in a global solution. As a result, they propose a distance correction for discrete demand in the spirit of Eyster et al. (1973)'s hyperboloid approximation. With this distance correction, discrete demand results in smoother objective functions. However, this distance approximation requires a parametric estimation for the correction factor.

Carrizosa et. al (1998) study the single facility Weber location problem with uniform regional demands. In their approach, to reduce the computational cost of evaluating the objective function, they approximate the rectangular demand regions, first with disks and then with triangles. Chen (2001) considers single facility Weber location problems where demand is in the form of circular areas each with uniform demand density. The solution procedure is an adaptation of the Weiszfeld's method with the evaluation of three scenarios, where location is inside, on or outside the circle.

2.3 Location-Allocation Problems

2.3.1 Location-Allocation problems with Discrete Demand

This problem is first introduced by Cooper (1963). This problem is a much complicated than the multi-facility location problem due to the need of concurrently deciding on the allocation decisions (combinatorial component) and location decisions (nonlinear component) and is shown to be NP-hard by Megiddo and Supowit (1984). Cooper (1964) proposed a sequential location-allocation procedure (SLA) for Euclidean metric planar problems which is in the same spirit as Maranzana (1964)'s approach for discrete p-median problems.

One dimensional version of the location-allocation problem is solved via dynamic programming by Love (1976). Denardo et al (1982) provide an interesting property for the location-allocation problem on a line with examples. This property, called interleaving property, stipulates that whenever a facility is removed from a solution, other facilities shift toward the location of the one removed, but not farther toward it than the original location of the adjacent facility.

Manhattan-metric Case It can be shown that for the Manhattan-metric case optimal locations of this problem will be at the grid points of vertical and horizontal lines passing through customer locations Francis and White (1998). Kuenne and Soland (1972) present a branch and bound procedure based on constructive assignment of customers to facilities. Love and Morris (1975) present a two-stage procedure which includes a set reduction step to reduce the set of optimal locations so that problem transforms into a p-median on a graph. Accordingly, the second step is the solution of this p-median problem. Shetty and Sherali (1977) provide a cutting plane approach for the multi-commodity formulation. Love and Juel (1982) propose five heuristics based on the exchange of the allocation decisions. Sherali et al. (1994) study capacitated version of the location-allocation problem. Based on a stronger property of the optimal locations due to Wendell and Hurter (1973), i.e., candidate grid points should be in the convex hull of demand points. Sherali et al. (1994) develop an equivalent discrete formulation. Resulting formulation is a bilinear mixed-integer program which is solved by a special linearization technique.

Euclidean-metric case: In the case of two-facility problems, Francis and White (1998), Ostresh (1975), Drezner (1984) and O'Kelly (1986) are examples of studies which use the problem's convex-hull property where two optimal subsets are separated by a line. This property, due to Ostresh (1975), stipulates that when there are n demand points, there are n(n-1)/2 possible subset partitions one of which would correspond to the optimal solution. As Ostresh (1975) and Drezner (1984) note, extension of this property to more than two subsets is too complex. Rosing (1992) further extends this convex hull property by generating all feasible convex hulls (i.e. subsets) which form the search space for the optimal allocation. Then he formulates a set covering model on these generated subsets. This method is capable of solving small sized problems optimally. Aykin and Brown (1992) propose a variant of the SLA approach for the LAP with interacting new facilities.

Cooper (1964, 1967) propose a sequential location-allocation method (SLA) which alternates between solving single facility location problems and a transportation problem by fixing the allocation decisions and location decisions, respectively. Love and Juel (1982) show that this problem can be expressed as concave minimization problem and present five heuristic methods. Kuenne and Soland (1972) propose a branch-and-bound algorithm which optimally solves small sized problems based on geometric lower bounds. Chen (1983) approximates the nearest-neighbor assignment with special exponents of distances and then provides a Quasi-Newton solution procedure for solving it. Multi-start together with lower bounding for location problems is suggested to overcome the local optimality. Murtagh and Niwattisyawong (1982) propose a simultaneous solution approach by relaxing the binary constraints on the allocation decisions. They use the MINOS nonlinear optimization package to solve iteratively by fixing the allocation variables whenever they are 0 or 1. They continue until no free allocation variable is left. Bongartz et al. (1994) develop an approach where, as in Murtagh and Niwattisyawong (1982), the binary constraints on allocation variables are relaxed. This method solves for location and allocation decisions simultaneously by using the sufficient conditions of the relaxed problem. Their algorithm is based on active set methods and orthogonal projections, and exhibits convergence rate between linear and quadratic. This approach is different than Murtagh and Niwattisyawong (1982) in that it exploits the special structure of the problem and uses Newton update rather than a Quasi-Newton update. Brimberg and Love (1998) has shown that two dimensional location-allocation problems can be solved with a dynamic programming approach if the demand data set possesses a one-dimensional intrinsic property.

Brimberg and Mladenovic (1996a) introduce Tabu search (TS) to one of the heuristics presented in Love and Juel (1982). Variable neighborhood search (VNS) framework, which allows a systematic way of exploring different regions of the search space, is adopted for the location-allocation problem in Brimberg and Mladenovic (1996b). Houck et al. (1996) present a Genetic algorithm (GA) implementation and comparison with the random-start and two-opt heuristic (H4) in Love and Juel (1982). Salhi and Gamal (2005) provide another genetic algorithm application. A combination of the GA with TS is presented in Houck et al. (2006). Chen et al. (1998) consider classical location-allocation problem and two other variants: one with existing facility interaction and other with a constant distance measure beyond as a threshold distance. They formulate the problem as a d.c. programming problem (maximization of differences of convex functions over a convex feasible set). They use an outer approximation for the resulting concave minimization problem. Hansen et al. (1998) introduce a p-median heuristic. P-median heuristic first solves the p-median problem over demand locations then single-facility Weber problem for the allocation decisions obtained in the first step. Brimberg et al. (2000) present an extensive empirical comparison of SLA, Bongartz et al. (1994)'s projection method, GA, TS, VNS, p-median heuristics and newly introduced add/drop heuristics. In their comparison they implement the SLA approach with 100 random starting points and then use the computational time to limit time allocated for other methods. Their results indicate that

VNS and p-median are better than the rest with respect to averaged gap from the best known solutions. Gamal and Salhi (2001) provide three constructive heuristics: multi-start SLA, furthest-distance and perturbation method plus SLA. Essentially the last one is identical to the p-median heuristic in Hansen et al. (1998) except the algorithm for solving p-median. Their results, in line with Brimberg et al., that for small number of facilities (n<20) multi-start SLA is better, but p-median heuristic performs better for larger number of facilities. Levin and Ben-Israel (2004) modify the SLA method by replacing Weiszfeld's single facility location step with author's Newton-bracketing procedure proposed in Levin and Ben-Israel (2002). Sherali et al. (2002) present a global solution procedure for capacitated general metric location-allocation problems. Their approach is based on branch-and-bound technique with two specialized lower bounding approaches. They report on n=30 and n=50 demand points which the method could solve optimal and heuristic branching procedure, respectively.

Taillard (2003) proposes three heuristic solution approaches for Squared Euclidean-metric problems. First heuristic, candidate list search, is a greedy heuristic based on perturbation of a solution and application of one step SLA. Second heuristic and third heuristics are based on choosing and solving subproblems using the first heuristic. Second heuristic, called local optimization, chooses a random facility and few of its closest facilities to create a subproblem. Third heuristic differs from second in the sense that it considers a number of facilities at the same time hence creates a number of subproblems. The issue of assigning optimal number facilities to each subproblem is determined by a dynamic programming. Based on computational study results first heuristic is recommended for moderate number of facilities whereas second and third are better performing for large number of facilities. In this study, the author also points out the efficacy of the p-median heuristic in Hansen et al. (1998) and refers to earlier work as follows:

"We do not consider methods such as those of Bongartz et al. (1994) which are too slow and produce too poor solutions or those of Chen (1983) or Murtagh and Niwattisyawong (1982) which are not competitive according to Bongartz et al. (1994)."

2.3.2 Location-Allocation problems with Continuous Demand

On a line Eaton and Lipsey (1975) analyze the cases where demand is represented with uniform and nonuniform density functions. They present necessary conditions for the equilibrium of allocation decision between two firms. They conclude that nonuniform demand solutions notably differ from the uniform case both in terms of service regions size and facility locations. Drezner and Wesolowsky (1996) consider a different problem where customers are traveling to the least expensive facility. Price at the facilities depends on the demand volume served. They provide first-order conditions for the equilibrium and propose three solution procedures.

Leamer (1968) considers three alternative shapes for the market region with uniform demand density. The solution procedure employed is a heuristic based on perturbation of the locations. Maruchek and Aly (1981) consider the case where demand is uniform randomly distributed over rectangular areas and propose a branch and bound solution. Cavalier and Sherali (1986) introduce the formulation of area demand location allocation problem which is a multi-facility location-allocation problem with uniform demand over convex polygonal areas. For single facility problem they develop a heuristic based on the triangulation of the demand region. For multi-facility case, they use the Cooper (1964)'s sequential location-allocation approach. Drezner (1986) considers the problem with circular demand areas and Squared Euclidean-metric where demand is uniformly distributed. Fekete et al. (2005) consider multi-facility location-allocation problem for the Manhattan-metric case. They provide a polynomial time algorithm for single-facility and prove the NP-hardness for multi-facility case.

2.4 Continuous Approximation approach for Location-Allocation problems

The idea of Continuous (a.k.a. continuum) Approximation (CA) is to convert finite-dimensional problems with a large number of variables into problems involving continuous functions. In the context of logistics, Daganzo (1991) builds on the work of Newell (1973) and addresses different transportation problems by providing a number CA models for different system settings. A detailed taxonomy of CA applications in transportation can be found in Langevin et al. (1996). In the economics literature, continuous models for competitive location problems are also used to obtain very neat analytical results. However, due to modeling complexity, results are only available for either uniform spatial demand distributions or in a single dimensional setting. Eaton and Lipsey (1975) and Beckman and Thisse (1986) are very good samples of this literature. From the location point of view, Geoffrion (1979) is the first to propose a continuous approximation model for a warehouse location problem. In this model, parameters including demand are constant across the region; hence, a unique optimal service region size is determined by minimizing the total of fixed cost, linear cost of operating, and inbound and outbound transportation cost densities. Erlenkotter (1989) presents a General Optimal Market Area (GOMA) model for uniform demand distribution. GOMA model trades off facility operating and outbound transportation costs, but ignores fixed facility and inbound transportation costs. Campbell (1992) studies location of multiple terminals to serve a uniformly distributed demand. This model assumes that terminals are supplied from a single source and transshipments between terminals allowed. They use the continuous approximation approach to analytically determine the optimal terminal sizes. They conclude that location decisions are not significant as long as allocation decisions are made optimally, i.e., nearest allocation may not be optimal. Rosenfield et al. (1992) implement the GOMA model to determine optimal service territory size for the United States Portal Service. Webster and Gupta (1995) extend the GOMA model by considering uncertain demand in a dynamic environment. Erlebacher and Meller (2000) propose a heuristic solution technique for inventory-location model which uses CA for the transportation costs. In their grid-based customer representation, they point at the importance of spatial distributions (demand in this case) on performance. They report a 7-39% accuracy range with their heuristic for a 600 customer and nine distribution center test problem. Recently, Rutten et al. (2001) refined the GOMA model by more precise modeling of fixed costs, of inventory costs and of transport cost. Their model differs from GOMA in that they assume square regions and a finite market size which allows them to specify number of depots to be located as the decision variable. Their results indicate that effect of considering market boundaries together with more accurate cost calculations leads to notably different results than GOMA model. Dasci and Verter (2001) generalize the GOMA model by allowing parameters to vary spatially, and then applied it to the facility location problem with capacity acquisition. Later, Dasci and Laporte (2004) extend GOMA to stochastic demand case. Dasci and Verter (2005) advance the GOMA model to analyze the plant focus strategies.

The main result of all GOMA models is an analytical optimal service region size given the parameters. A complete implementation requires allocating (or districting) the market region based on this formula. Dasci (2001) recommends using a bi-variate step function fitting for continuous optimal service region size. Ouyang and Daganzo (2006) propose a heuristic procedure which iterates discs that are sized according to the optimal service region formula. Another districting method is proposed by Novaes et al. (2000) for the continuous approximation approach to determine the type and size of a homogenous fleet for distribution from a single depot. Their method is based on allocating radial-ring shaped service regions to vehicles. Accordingly, this shape assumption leads to an explicit optimization problem which is solved by genetic algorithm.

2.5 Voronoi Diagram approach for Location-Allocation problems

Voronoi diagram is another approach towards solving planar location optimization problems when there are large number of demand points. Voronoi diagrams are formed by associating all points in a given set (i.e. on a plane) to the closest members of another set which is finite and presumably smaller. As a result of this nearest-neighbor assignment, polygonal tessellations are formed. Okabe et al. (2000) and Okabe and Suzuki (1997) provide excellent surveys on different applications of Voronoi diagrams in locational optimization problems. Iri et al. (1983) propose a gradient based search over a non-uniform customer distribution and applied to the optimal configuration of public mail boxes in Koganei, Japan. Their search is based on displacement of facilities in the optimal direction, namely towards the centroidal location for the squared Euclidian metric used in the study. Okabe et al. (1998) look at the location of hierarchical facilities where service provided by facilities at a hierarchical level encompasses all the services provided at lower levels. Their problem solves for the optimal levels of the hierarchy and the optimal system configuration of each level.

2.6 Conclusions

From the review of this literature it is apparent that there is scarcity of studies (both the pure-location and location-allocation problems) with continuous demand. Problems with discrete demand representation are challenged with the algorithmic limitations, i.e., lack of exact procedures for handling large size problems Taillard (2003). Furthermore, the non-convex nature of the location-allocation problems result in many local minima for discrete cases, thus global optimization techniques in the form of traditional and meta-heuristics appear to be promising (Taillard 2003, Brimberg et al. 2000). On the other hand, the continuous demand assumption smoothes the objective function by removing much of these local minimums hence reduces the need for heuristic search techniques (Drezner 1997). In comparison, when the demand data is dense, aggregation techniques bring about aggregation errors and whereas continuous approximation reduces this error. Therefore, continuous demand not only reduces the error, but also smoothes the objective function so that it exhibits less probability of getting stuck at local solutions.

On the other hand, continuous approximation, an approach which assumes continuous demand density, is a promising method, but suffers from the slowly varying assumption which in turn brings in solution errors. There also seems to be a lack of a consistent allocation/districting methodology to implement the nice analytical solutions obtained from continuous approximation. Another approach which is promising is by Iri et al. (1993) where the problem is solved in the location variable space with continuous demand. The only limitation of this approach is the reliance on the location decisions. Accordingly, when the costs are demand dependent, or when there are constraints on the demand to be served (either distance constraints or capacity constraints on the total demand served), then location based approach would not be extendable. So there is a need for models and methods which could benefit from the favorable aspects of using continuous demand data, but at the same time versatile enough to allow various problem variations that discrete demand models could easily account for.

Chapter 3

Basic Model on Single Dimension

3.1 Introduction

In this chapter, we develop a dynamic programming formulation on a line for a fixed-charged continuous location-allocation problem with capacity acquisition, which is similar to the discrete model proposed in Verter and Dincer (1995). This problem is a representative form of the many supply chain network design models found in the literature. Our model differs from Verter and Dincer (1995) in that we consider continuous facility locations and continuous demand functions. As in Verter and Dincer (1995), we also assume that any capacity can be acquired without any constraint. Single sourcing, assigning all of the demand at each customer to the closest facility, is therefore a direct result of this acquirable capacity assumption. Single sourcing assumption not only makes the analysis easier but also is the most preferred method in practice as it simplifies the management of distribution.

Our continuous location-allocation problem (also called service region districting - location problem) formulation is based on prioritizing the allocation decisions. In our approach, we determine the customer allocation decisions by districting the market into service regions in which we assume the facilities are optimally located. In most of the discrete and planar location-allocation models, optimal location and allocation decisions are jointly determined. Continuous model formulations, on the other hand, prioritize the location decisions while allocation decisions are determined optimally given the location decisions. For a better exposition of this difference, let's consider the following optimization problem where the location decisions (x) and allocation decisions (y) are jointly determined. This formulation is akin to the discrete and planar models suggested for the location-allocation problems.

$$\begin{aligned} &\min_{x,y} f(x,y) \\ &s.t. \\ &x \in X, y \in Y \end{aligned} \tag{1}$$

In the continuous models, the location prioritized formulation is of the form:

$$\begin{aligned} &\min_{x} f(x,y^*(x)) \\ &s.t. \\ &x \in X \end{aligned} \tag{2}$$

$y^*(x)$ is the optimal allocation decisions implied by the location decisions x. One can easily observe that this formulation has a smaller decision space compared to the formulation in (1). Whereas this reduction of the search space facilitates the solution, the optimal allocation decisions (y^*) further complicates the problem. In our formulation, we instead prioritize the allocation decisions. $x^*(y)$ is the optimal location decisions implied by the allocation decisions y.

$$\begin{aligned} &\min_{y} f(x^*(y),y) \\ &s.t. \\ &y \in Y \end{aligned} \tag{3}$$

In this chapter, we consider the fixed-charged continuous location-allocation problem with capacity acquisition on a simplified domain, namely on a line. It constitutes not only the basic construct for the two dimensional problems but also an easy to explain setting without losing the reader in technicalities and complex notation arising from additional dimension. In Section 3.2., we first present the traditional formulation of the problem. Then, in Section 3.3, we propose an alternative dynamic programming model on a line and derive some analytical properties of the optimal solution. In the last section, Section 3.4., we present two alternative solution methodologies together with examples. These two methods are constructive shooting algorithm and steepest-descent algorithm based on allocation decision iterations. Although, the demand density function is assumed to be linear throughout this chapter, our results are applicable for nonlinear demand cases as well so long as the demand density is Lipschitz continuous. Only exception to this generalization is the pseudo-convexity results, Propositions 3.1 and 3.2, which are based on the linear demand density assumption.

3.2 Basic Model- Allocation Variable Space

3.2.1 Description of the Decision Variables and Parameters

The purpose of constructing the model for the allocation problem on a line is two fold. Firstly, it is easier to derive analytical properties characterizing the optimal solution. Secondly, it forms a basis for the extension to the planar model as described in the planar model section. A feasible solution for the allocation problem defined on a line is a set of line segments which are disjoint and cover the market region completely Figure 3.1. exhibits a feasible solution

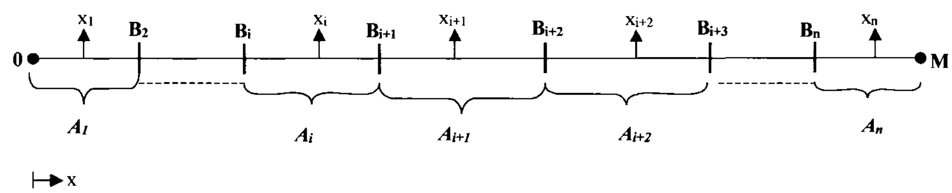

Figure 3.1: A feasible solution to the service region districting and location problem on a line.

for the problem on a single dimensional market which is defined as the line segment $[0, M]$ where the origin(0) is any reference point. $\mathcal{A}_i$'s are the service regions which disjointedly covers the market $[0, M]$. The idea of service regions is that any customer demand in $\mathcal{A}_i$ will be served from the facility located in service region i at coordinate x_i.

First, we colloquially define all problem parameters and decision variables, and then in the following section, we develop the mathematical relations between them and substitute the equivalents in the model formulation when necessary.

Decision Variables

n : number of facilities (service regions)

A_i : area of service region i (i.e. $\mathrm{A}_i = |\mathcal{A}_i|$)

B_i : coordinate of boundary between service regions i and $i-1$

Auxiliary Variables

x_i : coordinate of the facility in service region i

w_i : total demand in service region $\mathcal{A}_i$

$\overline{x}_i$: coordinate where the average density in $\mathcal{A}_i$ is observed $\left(\text{i.e. } D(\overline{x}_i) = \frac{\int_{\mathcal{A}_i} D(x) \ dx}{A_i}\right)$

Problem Parameters

$D(x)$	: demand density at x$\in M$ (items/mile) $(= u + vx)$
F	: fixed cost of opening a plant
$f(x, w)$	: capacity acquisition cost of opening a facility of size w at x
$g(x, w)$	: total transportation cost for serving demand volume w from a facility located at x
f	: fixed cost component of capacity acquisition cost
a	: unit capacity acquisition cost
c	: per unit-mile distribution cost
$\overline{d}_i(x_i)$	: average customer travel distance in $\mathcal{A}_i$ when facility is located at x_i for direct-shipment case.
$\eta(x, x_i)$	: distance from the facility at x_i to x

3.2.2 Cost Factors and the Model

Objective of the problem is to minimize total cost of meeting all demand which includes minimization of total fixed costs of opening facilities, capacity acquisition and transportation costs. We now give explicit forms of these costs based on service districting(allocation) variables, namely A_i, $i = 1..n$.

Capacity Costs

We assume that capacity acquisition cost is a fixed charged linear function of the demand served as shown in Figure 3.2. It also serves as an approximation to the generalized nonlinear monotone decreasing per-unit capacity costs. The healthcare capacity is a good example where there is an initial fixed cost of "ready-to-serve" capacity and a marginal cost component derived from the rendered services (Lindelöw and Wagstaff 2003). Consideration of explicitly

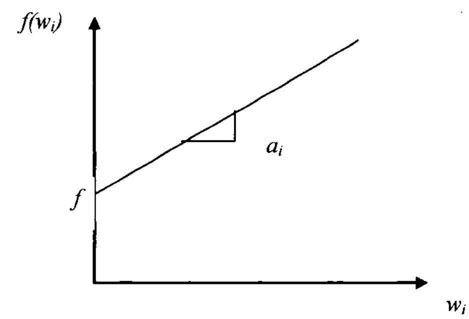

Figure 3.2: Fixed charged variable capacity acquisition cost

nonlinear capacity costs is proposed as a future research in the later chapters.

Formulation of the fixed-charged linear capacity cost is as follows:

$$f(w_i) = f + a_i w_i = f + a_i(\mathrm{A}_i D(\overline{x}_i)) \tag{4}$$

Since the demand is linear, we have replaced w_i with the average demand multiplied with the area. Observe that whereas in the uniform demand case capacity is solely determined by the size of service region, in the linear demand case location of the service region in the market region is also a factor.

Distribution Costs

We assume direct shipping distribution systems where each customer gets a reserved dispatch and transportation costs are charged on a per unit-mile basis. Distribution strategy is one-to-many meaning that each facility has its dedicated territory that no other facility could supply customers in another service region. It is also assumed that trucks and customers are identical in their cost and time factors. Under these assumptions, total transportation cost in service region $\mathcal{A}_i$ becomes,

$$g(x_i, w_i) = c_i \overline{d}_i(x_i) w_i = c_i \overline{d}_i \left(\mathrm{A}_i D(\overline{x}_i) \right) \tag{5}$$

In the continuous approximation to transportation literature, average travel distance, $\overline{d_i}(x_i)$, has been popularly approximated by the area size. In these models demand is assumed to be uniform and facility locations are static (i.e. known before and kept constant). Almost all of these models use a coefficient that reflects the effect of number of routes (in peddling case), shape of the distribution area, location of the facility with respect to the transportation area (inside or outside) and distance metric used in calculating travel distances. For single dimensional analysis, shape naturally does not have any affect and distances are same for $L_1(Manhattan)$ and $L_2(Euclidian)$ metrics. However, if the demand density uniformity and fixed location assumptions are relaxed, it is intuitive to conjecture that $\overline{d_i}$ would be some function of the demand distribution and area. In general average travel distance, $\overline{d_i}(x_i)$, for service region $\mathcal{A}_i$ could be calculated as follows;

$$\overline{d}_i(x_i) = \frac{Total\ Distance\ Traveled\ (x_i)}{Total\ Demand} = \frac{\int_{\mathcal{A}_i} D(x)\ \eta(x, x_i)\ \ dx}{\int_{\mathcal{A}_i} D(x)\ dx} \tag{6}$$

Total distance traveled,for the representation in Figure.3.3 is as follows;

$$\begin{aligned} Total\ Distance\ Traveled\ (x_i) \quad &= \quad \int_{\mathrm{B}_i}^{x_i} (u + v\,x)\,(x_i - x)\,dx & (7) \\ &+ \int_{x_i}^{\mathrm{B}_i+\mathrm{A}_i} (u + v\,x)\,(x - x_i)\,dx & (8) \end{aligned}$$

Given the districting solution, the location of the facility, x_i^* is the one that minimizes total transportation cost. In the case of *Manhattan* metric, this x_i^* is the well-known *median* location of $\mathcal{A}_i$ with respect to demand density function. x_i^* can be derived from the first-order optimality condition after taking the derivative of (7).

$$x_i^* = \frac{1}{4}\frac{-4\,u + 2\,\sqrt{4\,u^2 + 4\,v^2\,\mathrm{B}_i\,\mathrm{A}_i + 4\,v^2\,\mathrm{B}_i{}^2 + 4\,u\,v\,\mathrm{A}_i + 8\,u\,v\,\mathrm{B}_i + 2\,v^2\,\mathrm{A}_i^2}}{v} \tag{9}$$

Note that (9) is the optimal location for Euclidean metric (L_2) case as well, since Euclidean and Manhattan metrics are identical in single dimension.[1] A more clear to remember expression for the x_i^* could be expressed in terms of the demand densities at the beginning and end points of the service region $\mathcal{A}_i$.

$$D(x_i^*) = \left(\frac{D(B_i)^2 + D(B_i + A_i)^2}{2}\right)^{\frac{1}{2}} \tag{10}$$

In the *Manhattan* metric, when the facility is located at x_i^*,the average distance traveled, $\overline{d_i}(x_i^*)$, from equation (6):

$$\overline{d_i}(x_i^*) = K(T_i)A_i$$

$$K(T_i) = \frac{\left(\begin{array}{c} 2T_i^3 - (\sqrt{2T_i^2 + 4T_i + 4} - 6)T_i^2 \\ -(2\sqrt{2T_i^2 + 4T + 4} - 6)T_i - 2\sqrt{2T_i^2 + 4T_i + 4} + 4 \end{array}\right)}{3T_i^2(T_i + 2)} \tag{11}$$

[1]In the case of squared Euclidian distance metric, x_i^* would be at the *centroid* (*center of mass*) of $\mathcal{A}_i$ with respect to $D(x)$. $x_i^* = \frac{A_i(2vA_i+3u+3vB_i)}{3(vA_i+2u+2vB_i)}$

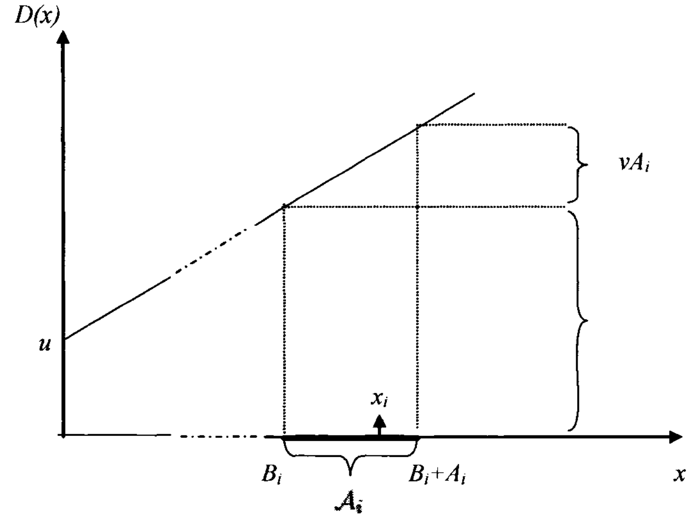

Figure 3.3: An example representation of the service region $\mathcal{A}_i$ with linear demand density function in single dimension.

$$T_i = \mathcal{T}(\mathrm{A}_i,\mathrm{B}_i) = \frac{v\mathrm{A}_i}{u + v\mathrm{B}_i} = \frac{v\mathrm{A}_i}{D(\mathrm{B}_i)} \tag{12}$$

T_i : demand density elevation measure in $\mathcal{A}_i$, i.e., the ratio of the demand density level elevation to the lowest level of demand.

The dependence of $\overline{d}_i(x_i^*)$ on T_i is a special case due to the linearity of demand and is considerably more involved for nonlinear demand cases. Also note that there is an area A_i multiplier in the end, so in addition to the curvature properties of the demand density function, it also depends on the area size of service region. $K(T_i)$ is an average travel distance coefficient which depends on the shape of the $\mathcal{A}_i$, distance metric used and demand density distribution. The continuous approximation literature (Daganzo 1991, Erlenkotter 1989, Geoffrion 1979, Dasci and Verter 2001, Rutten et al. 2001), given the shape and metric, assumes $K(T)$ as a constant value, but when we relax the uniform demand assumption, it is no longer a constant.

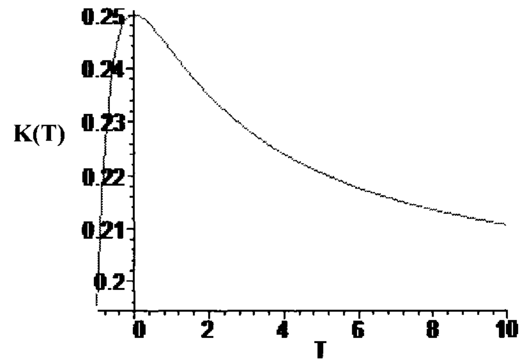

Figure 3.4: $K(T)$ for both linearly increasing (for $T \geq 0$) and linearly decreasing (for $T \leq 0$) demand density functions.

Figure 3.4 demonstrates the behavior of $K(T)$ for linearly increasing and decreasing demand density functions. Note that as the demand density function slope $v \rightarrow 0$ then $T \rightarrow 0$ and $K(T)$ parameters both converge to 0.25 which is the correction factor for average distance traveled in single dimension with facility location is at the center of service region. Asymptotically, when $T \rightarrow +\infty$ and $T \rightarrow -1$ (i.e. $v \rightarrow +\infty$ and $v \rightarrow -\infty$), then $K(T) \rightarrow 0.195$.

With these results, the total transportation cost in (5) is as follows:

$$g(x_i^*, w_i) = c_i K(T_i) \mathrm{A}_i \left(\mathrm{A}_i D(\overline{x}_i) \right) \tag{13}$$

In the uniform demand case $K(T_i)$ is dependent on the shape of the service region and the distance metric used, whereas in the linear demand case it also depends on the size of the service region, A_i. When uniform demand density is assumed, the error in the distribution cost could be as high as 28%.[2] Moreover, this uniformity assumption would also displace the optimal facility location (with a limiting value of 20.7% for the model herein).[2] These errors

[2] Asymptotically when $v \rightarrow +\infty$ and $v \rightarrow -\infty$, assuming uniform demand, i.e. $K(T) =$ 0.25 and $\mathrm{x}_i^* = \mathrm{B}_i + \frac{\mathrm{A}_i}{2}$, would bring 28% distribution cost and 20.7% displacement error for the optimal location based on (11) and (9), respectively.

are for a single service region only. When many service regions are considered in tandem, this error of uniform demand density assumption propagates additively for linear demand density.

We now combine the cost terms and express the objective function of our allocation based service districting-location and capacity acquisition problem:

$$\min_{n \in Z^+, \mathrm{A}_i} TC = \sum_{i=1}^{n} TC_i(\mathrm{A}_i) = \sum_{i=1}^{n} \left[F + f + a\mathrm{A}_i D(\overline{x}_i) + cK(T_i)\mathrm{A}_i^2 D(\overline{x}_i)\right] \quad (14)$$

Any feasible solution $(n, \mathrm{A}_{i=1..n})$ to our problem should cover the market area (i.e. $\cup \mathcal{A}_i = \mathcal{M}$ where $\mathcal{A}_i$ are disjoint). Auxiliary variables in (14) are $T_i = v\mathrm{A}_i/u + v\sum_{j=1}^{i-1}\mathrm{A}_j$ and $\overline{x}_i = \sum_{j=1}^{i-1}\mathrm{A}_j + \frac{\mathrm{A}_i}{2}$. With (14), our problem is a mixed integer non-linear function hence a non-convex programming model defined over A_i and n. Non-convexity is primarily due to the discrete nature of feasible region as necessitated by the number of service regions (n). When the number of facilities (n) is given, this problem becomes a pseudo-convex problem as it is shown in the following section. In the case of non-linear demand density, this pseudo-convexity depends on the structure of the demand density function (i.e. convex when demand density is strictly-convex, concave when demand density is strictly-concave).

3.3 Alternative Model- Dynamic Programming Model in the Allocation Variable Space

We now give a mathematical programming formulation for our allocation problem for the *Manhattan* metric and similar derivations for the *Euclidian Squared* metric is straightforward. We replace the auxiliary variables $(T_i, \overline{x}_i)$

with their equivalents using boundary variables $B_i(=\sum_{j=1}^{i-1} A_j)$. This formulation is provided below. For notational simplicity, we use total cost mapping as before $(TC(\cdot))$ but this time with two variables (B_i, A_i).

*Problem P*1

$$\min_{n,A_i,B_i} TC = \sum_{i=1}^{n} TC_i(B_i, A_i) = \sum_{i=1}^{n} \left[\begin{array}{c} F + f + a\mathrm{A}_i D(B_i + \frac{\mathrm{A}_i}{2}) + \\ cK(\frac{v\mathrm{A}_i}{u+vB_i})\mathrm{A}_i^2 D(B_i + \frac{\mathrm{A}_i}{2}) \end{array} \right]$$

$s.t.$

$$B_{i+1} = B_i + \mathrm{A}_i \tag{15}$$

$$B_1 = 0,\ B_{n+1} = M,\ \mathrm{A}_i \geq 0,\ \mathrm{B}_i \geq 0,\ n \geq 0 \textit{ and discrete} \tag{16}$$

Since $P1$ has an additive and separable objective function with linear constraints– each relating boundary decision variable B_{i+1} to the previous boundary variable (B_i) and service area variable (A_i), this formulation is a good candidate for the dynamic programming formulation for a given n. Denoting B_i's as the state variables, A_i's as the control variables, (15) as the system equation, we obtain the cost-to-go function as follows:

$$V(B_i, i) = \underset{B_{i+1}\geq 0, A_i \geq 0, \lambda_i}{minimize} \{TC_i(B_i, A_i) + \lambda_i(B_{i+1} - B_i - A_i) + V(B_{i+1}, i+1)\} \tag{17}$$

$V(B_i, i)$: the cost of optimal allocation decisions starting from B_i and i^{th} facility

First two terms in (17) represents the Lagrange function of the optimization problem in service region i and λ_i stands for the Lagrange multiplier for the boundary condition of the service region. One way to interpret this formulation is to think of it as an event-driven system where every facility has to select its service region based on the previous allocation decisions. The basic idea of (17) is to start the problem of districting at an area i onwards (i.e. $[B_i, M]$). If we move to the next area $i+1$ and restart the problem again, we will have the same service area districting solution for the areas $i+1$ onward (i.e. $[B_{i+1}, M]$). In other words, if the solution to the i^{th} service districting problem is (A_j^*, B_j^*), $j = i, i+1..., n$ then the solution to the $(i+1)^{th}$service districting problem is (A_j^*, B_j^*), $j = i+1..., n$. Note that for any given service region i, B_{i+1} is also a control variable.

The cost-to-go function in (17), $V(B_i, i)$, satisfies the principle of optimality such that optimal trajectories of choice variables (A_i) are as functions of state variables (B_i). Since Bellman's equation in (17) is a contraction mapping, for which a fixed point theorem exist, the existence of a solution is guaranteed. The uniqueness of this solution depends on the convexity properties of the Hamiltonian equation in (18).

$$H(B_i, A_i, \lambda_i) = TC_i(B_i, A_i) + \lambda_i \mathrm{A}_i \tag{18}$$

If (18) is convex then the contraction mapping implied by Bellman's equation would preserve its convexity. However, (18) is not convex in (A_i, B_i, λ_i), thus we show the unimodularity of our problem through its weak convexity. In what follows, we first derive the necessary conditions for the optimal service region districting problem and then prove that our problem is a pseudo-convex problem; thus, necessary conditions are also sufficient.

3.3.1 Necessary and Sufficient Conditions for Optimality

In this section we first derive the necessary conditions of the Minimum Principle through the Bellman equation in (17) and then present the sufficiency of these conditions for an optimum solution. Note that the decision variables in (17) are A_i, B_{i+1} and λ_i whereas B_i serves as a parameter in the ith area's problem. An optimal solution to the (17) is a triplet $(A_i, B_{i+1}, \lambda_i)$ for $i = 1...n$ and must satisfy the following first order conditions for any given n:

$$\frac{\partial V(B_i,i)}{\partial A_i} = 0: \quad \frac{\partial TC_i(B_i,A_i)}{\partial A_i} = \lambda_i \tag{19}$$

$$\frac{\partial V(B_i,i)}{\partial B_{i+1}} = 0: \quad \frac{dV(B_{i+1},i+1)}{dB_{i+1}} = -\lambda_i \tag{20}$$

$$\frac{\partial V(B_i,i)}{\partial \lambda_i} = 0: \quad B_{i+1} = B_i + A_i \tag{21}$$

$$B_{i+1}, A_i \geq 0 \; i = 1...n \tag{22}$$

We are assuming that the solution $(A_i, B_{i+1}, \lambda_i)$ for $i = 1...n$, is an internal solution with respect to the non-negativity set 22, i.e. $A_i, B_{i+1} > 0$. This is not a strict assumption since given an initial boundary $B_1 \geq 0$, B_{i+1} would be positive except $\sum\limits_{j<i} A_i = 0$. However, since we derive these optimality conditions for a given n, then having some m number of the A_i's at zero would mean we have an optimal solution for $n - m$ service regions. In addition to (19),(20) and (21) we have the following condition from the envelope theorem (Kimball 1952).

$$\frac{dV(B_i,i)}{dB_i} = \frac{\partial TC_i(B_i,A_i)}{\partial B_i} - \lambda_i \tag{23}$$

Through algebraic manipulation, we obtain the Euler equation (24) linking area sizes in adjacent service regions (Holzapfel 1986).

$$\frac{\partial TC_i(B_i, A_i)}{\partial A_i} = \frac{\partial TC_{i+1}(B_{i+1}, A_{i+1})}{\partial A_{i+1}} - \frac{\partial TC_{i+1}(B_{i+1}, A_{i+1})}{\partial B_{i+1}} \tag{24}$$

This result is the discrete version of the Minimum Principle, which is an $open - loop$ control approach to districting-location problem; the optimal control solutions $(A_i^*(B_i^*))$ are identified only at the optimal states (B_i^*). However, the dynamic programming approach in (17) is a *closed loop* approach, where the optimal control solutions are identified for every state condition (i.e. $\mathrm{A}_i^*(B_i)$). Therefore closed-form analytical results are rare with dynamic programming. Lastly, it is possible to show the following relation for λ_i.

$$\lambda_i = \frac{\partial TC_{i+1}(B_{i+1}, A_{i+1})}{\partial A_{i+1}} - \frac{\partial TC_{i+1}(B_{i+1}, A_{i+1})}{\partial B_{i+1}} \tag{25}$$

In order to derive (25), we write the envelope condition in (23) for $i + 1$ subtract (20) and substitute for λ_{i+1} in (19) for $i + 1$.

It is important to point out that above conditions are necessary but not sufficient conditions for an optimum solution, and for sufficiency we need to explore the convexity properties of $P1$. In general, sufficient optimality conditions are available for continuous dynamic systems only under some restrictive convexity assumptions (Leonard and Ngo 1992). However, as we show in Proposition 3.1. and 3.2., due to pseudo-convexity property, these conditions are also sufficient for an optimum. In further extension of our problem (i.e. nonlinear demand cases), we loose these weak convexity properties; hence, we are bounded with local optimization techniques.

Proposition 3.1.

For any given consecutive triplet of service regions $(\mathcal{A}_i, \mathcal{A}_{i+1}, \mathcal{A}_{i+2})$, $TC_{i,3}(B_i, A_i) = TC_i(B_i, A_i) + TC_{i+1}(B_{i+1}, A_{i+1}) + TC_{i+2}(B_{i+2}, A_{i+2})$, *s.t.* $B_{j+1} = B_j + A_j$ *for* $j = i, i+1, i+2\}$ *is a pseudo-convex function of* A_i *and* A_{i+1} *for a given* B_i *and* B_{i+3}.

Proof. *For brevity, we provide a description of the proof and an extended version can be found in the appendix. For* $v \geq 0$, *we first show that* $\partial TC_i(B_i, A_i)/\partial A_i$ *is a convex and increasing function. Also* $\partial TC_{i+1}(B_{i+1}, A_{i+1})/\partial A_i$ *is convex, increasing and* ≤ 0. *Further at* $A_i = 0$, *sum of these two functions is* ≤ 0, *and a convex increasing function of* A_i, *thus two area cost is unimodal in* A_i. *For three area case,* $\partial TC_{i+1}(B_{i+1}, A_{i+1})/\partial A_i + \partial TC_{i+2}(B_{i+2}, A_{i+2})/\partial A_i$ *is convex, increasing and* ≤ 0. *Thus three area costs is also unimodal in* A_i. *Since total cost of three area case is unimodal in* $A_i + A_{i+1}$, $TC_{i,3}(B_i, A_i)$ *is also unimodal in* A_i *and* A_{i+1}. *For* $v \leq 0$, *proof is done in an identical approach.* ■

Proposition 3.2.

Given the number of facilities, n, total cost function $TC = \sum_{i=1}^{n} TC_i(B_i, A_i)$ *is a pseudo-convex function of* (B_i, A_i) *for* $i = 1, 2, ..., n$.

Proof. *The minimal service districting problem would consist of two service regions and is a univariate optimization problem in one of the areas, thus is pseudo-convex. For the a triplet service region districting problem, previous proposition's results are sufficient for pseudo-convexity. In the case of four* *or more service regions, it follows from the previous proposition's results by induction. As we iterate i from n to 1, where $B_n = M$ and B_1 are given, the current area cost ($TC_i(A_i, B_i)$) is pseudo-convex in A_i and remaining areas' total cost ($TC_{i+1}(A_{i+1}, B_{i+1}) + TC_{i+2}(A_{i+2}, B_{i+2})$) is pseudo-convex and non-increasing function of A_i.* ∎

Therefore from the result of Proposition 3.2., our problem is unimodal and thus necessary conditions are also sufficient. Since Euler equation (24) is also a sufficient condition, we now present result for the physical interpretation of the optimal service districting-location solutions. With several steps of algebraic manipulations, one can obtain to the following relation.

$$\frac{\partial TC_i(A_i, B_i)}{\partial A_i} + \frac{dTC_{i+1}(A_{i+1}, B_{i+1})}{dA_i} = 0 \tag{26}$$

This is another version Euler equation and it states that, at the optimality, *the marginal rate of change of total cost in two neighboring areas cancel out each other.* Proposition 3.3. further elucidates this observation.

Proposition 3.3. *In the optimal solution, facility locations in every neighboring service region pair are equidistant from the shared boundary.*

$$A_i - (x_i - B_i) = d = x_{i+1} - B_{i+1} \tag{27}$$

Proof. *For brevity, we provide a description of the proof and an extended version can be found in the appendix. If we replace B_{i+1} with $B_{i+2} - A_i$ in (26), plug into the (14) and through several algebraic manipulation steps, we*

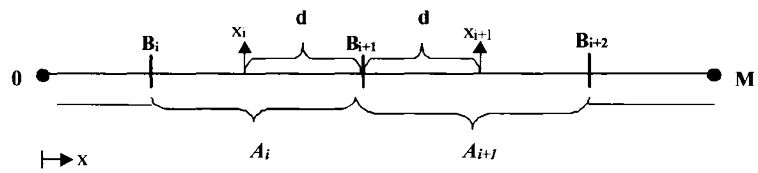

Figure 3.5: Equidistance property of the optimal solutions to service districting-location problem.

obtain this result. ■

Figure 3.5 depicts this proposition; facility locations at $\mathcal{A}_i$ and $\mathcal{A}_{i+1}$ are located d distance apart from the shared boundary B_{i+1}. This property is in a way a condensed expression of the optimality and feasibility conditions for the internal service regions of the market. However, for the peripheral market regions ($\mathcal{A}_1$ and $\mathcal{A}_n$), this condition, naturally, does not apply. For the peripheral service regions, we have the two boundary conditions as the constraints.

This result has an important implication on the manufacturer's districting and location decisions: customer's individual patronizing decisions concur with the manufacturer's optimal location and optimal assignment of customer's to the facilities. This implication is true for the service regions located interior, but in the peripheral service regions, $\mathcal{A}_1$ and $\mathcal{A}_n$, when the demand is increasing, customers to close to the market boundaries are travelling more (case $\mathcal{A}_1$) and less (case $\mathcal{A}_n$) than the rest in the respective service regions. This result also holds true for squared Euclidian metric. In our problem we regard the manufacturer as the only decision maker for the allocation-location decisions, whereas, in the economics literature, competitive location decisions are based on the assumption that customers patronize the closest facility and consequently each facility tries to maximize their market niche by their locations.

Our result in (27) together with the median location of x_i's establishes that our problem solution is equivalent to the competitive location decisions of the independent facilities. In the optimal solution, (27) correctly responds to the customer's minimum travel seeking behavior and the median location of x_i's models the competitiveness of i'th facility. If x_i is not at the median location, then it can increase its demand catchment by approaching in the increasing demand density direction and when it is at the median location this potential is zero. This concordance, however, does not hold true for the Squared Euclidian metric since optimal facility locations are at the centroid.

The primary use of this equidistance property is that it characterizes both the optimal location decisions and the optimal allocation decisions in the location-allocation problem. In the next section, we propose a shooting algorithm which is based on this equidistance property.

3.4 Solution Methodologies

There are three alternative solution approaches for the dynamic programming model in (17). First one is to construct a solution from scratch using the equidistance property. In contrast, the other two approaches are steepest-descent improvement based methods, which start with an arbitrary solution and iteratively improve allocation decisions in a descent direction. Difference between the later two methods is that, in the first algorithm we iterate allocation decisions one at a time, and, in the second algorithm, we jointly iterate allocation decisions.

3.4.1 Constructive Solution Approach

Shooting Algorithm

The optimality condition in (27) is similar to the two-point boundary value problems in differential calculus and therefore it can be solved from scratch via efficient numerical techniques such as shooting methods. Shooting algorithm is one of the most popular methods of solving boundary value problems of differential equations, especially when a closed form solution is not available. In this approach, we have a system for which there is a dynamic equation in the form a differential (or difference) equation of the system's state variable and two boundary values for each variable defining the origin and destination values of the system's state. The problem is that a closed form solution of the state variable is not available; therefore, as an alternative approach, an arbitrary change is triggered in the system's initial state (boundary value) and thereon followed until the end of the time horizon. If the initial trigger is not accurate then the ending state value differs from the boundary value. Therefore, the final state of the system and the path followed from the initial state is conditional on the initial change triggered– the more accurate the initial trigger, the closer the system gets to the boundary value at the end or time horizon. Since this briefly described shooting procedure aims to reach the targeted final boundary state by following a dynamic system equation, its performance is both conditional on the accuracy of system dynamics as well as the initial trigger.

In relation to our problem, the sequential tiling of the market area with service regions is analogous to the system dynamics described above. Therefore, adapting this general shooting method to solve our problem is rather straightforward. For a given value of n, we have two boundary conditions (B_1 and M) and starting with an initial estimate of A_1, one can incrementally tile the market region using the difference equation in (27). When n service regions are tiled, $B_{n+1} = B_1 + \sum_{i=1}^{n} \mathrm{A}_i$ is the implied boundary value for that starting value of A_1. Since $B_{n+1} = M$ is one of the model constraints, any solution where $|B_{n+1} - M| \neq 0$ is an infeasible solution. Therefore we revise the initial estimate of A_1 and repeat the same steps until we obtain a feasible solution which is also optimal.

Now we provide a formal shooting algorithm for our basic model:

Shooting Algorithm:

Step 1. Initialize the model parameters and variables

k : index for the number of service regions (i.e. $n^k < n^{\max}$)

i : index for the service regions (i.e. A_i and B_i for $i = 1...n^k$)

j : index for the feasibility iterations (i.e. $j^* = \{j|\ \epsilon_{BOUND} \geq |B^j_{n^k+1} - M|\}$

$\epsilon_{COST}, \epsilon_{BOUND}$: epsilon parameters for stopping decisions in the feasibility and optimality loops

A_1^j : j^{th} iteration estimate for the first service region size

B_1, M : boundary conditions

Set $k = 1, j = 1$, $n^k = 3$, $\mathrm{A}_1^{j=1} = \mathrm{A}_{\min}$ and $\mathrm{A}_1^{j=2} = \mathrm{A}_{\max}$

Step 2. Update the number of Service regions, n^k

Do While $\left(\frac{|TC(n^k) - TC(n^{k-1})|}{TC(n^{k-1})} \geq \epsilon_{COST}\right)$:

Repeat Step 3. and 4 for $n^k = (n^k - 1, n^k, n^k + 1)$

Calculate $n^{k+1} = \left\lfloor n^k - \frac{1}{2}\left(\frac{TC(n^k+1) - TC(n^k-1)}{TC(n^k+1) - 2TC(n^k) + TC(n^k-1)}\right)\right\rceil$, Return Step 2.

$k = k + 1$

Step 3. Update the first service region size, A_1^j

Do While ($|B^j_{n^k+1} - M| \geq \epsilon_{BOUND}$):

$j = j + 1$

If $j \geq 2$, then calculate $A_1^{j+1} = A_1^j - \frac{\left(B^j_{n^k+1}-M\right)\left(A_1^j - A_1^{j-1}\right)}{\left(B^j_{n^k+1} - B^{j-1}_{n^k+1}\right)}$.

Step 4. Tile the market region, $[\mathbf{B}_1, M]$

For $i = 1$ to n^k, REPEAT

Solve $\left(1 - \frac{(Z(T_i)-2)}{2T_i}\right) A_i^j = (A^j_{i+1}) \left(\frac{Z(T_{i+1})-2}{2T_{i+1}}\right)$ for A^j_{i+1},

where $Z(T_i) = \sqrt{2T_i^2 + 4T_i + 4}$ and $T_i = \frac{vA_i^j}{u+vB_i^j}$

$B^j_{i+1} = B^j_i + A^j_i$

$i = i + 1$

Return.

Step 5. Terminate with the solution $n^k, A^j_{i=1..n^k}$

The maximum and minimum value parameters for the first service region size, A_{max} and A_{min}, could be set at any value ranging from 0 to M. The algorithm operates on three different loops. First loop, Step 2, determines the number of service regions. Second loop in Step 3 determines an initial service region size (A_1). Third loop in Step 4 tiles the market region sequentially based on the number of service regions and initial service region size set in the previous loops. $TC(n^k)$ is the minimum cost solution for a given predetermined number of facilities (n^k).

In order to find the optimal number of service regions, we could start with an arbitrary small number and then iteratively increase by one until there is no improvement. Since this approach would be cumbersome, we choose a better approach in updating the number of service regions. More specifically, we approximate $TC(n^k)$ with a convex function and use a second-order root-finding algorithm (discrete version of the Newton-Raphson method) to update n^k. Hence, Steps 3 and 4 are repeated for three consecutive values of n (n^{k-1}, n^k, n^{k+1}). This is needed for the estimation of second derivative in the Newton-Raphson method. With this line search, we are assuming that total cost is unimodal in the number of service regions (n). We follow a similar logic for the update of A_1. The notation $\lfloor\rceil$ is used to denote the nearest integer function.

For the n^{k+1}, we use the standard approximation of the Newton Raphson method as explained next. We first write the Taylor series expansion of $TC(n^{k+1})$ around n^k.

$$TC(n^{k+1}) = TC(n^k) + TC'(n^k)(n^{k+1} - n^k) + \frac{(n^{k+1} - n^k)^2 TC''(n^k)}{2} + O((n^{k+1} - n^k)^3)$$

After taking derivative with respect to n^{k+1} and collecting terms, we obtain the following *iteration equation.*

$$n^{k+1} = n^k + \frac{TC'(n^k)}{TC''(n^k)}$$

Since we do not know the closed form functional expression of $TC(n)$, we approximate the first- and second-order derivatives with *centered difference equations.*

$$TC'(n^k) = \frac{TC(n^k + 1) - TC(n^k - 1)}{2}$$

$$TC''(n^k) = \frac{TC(n^k + 1) - 2TC(n^k) + TC(n^k - 1)}{1}$$

Thus the Newton-Raphson iteration function for n^k is,

$$n^{k+1} = n^k - \frac{1}{2}\left(\frac{TC(n^k+1) - TC(n^k-1)}{TC(n^k+1) - 2TC(n^k) + TC(n^k-1)}\right) \tag{28}$$

In updating the initial service region (A_1), we also use Newton's method, but this time for finding the root of the feasibility condition, i.e. market coverage. From the previous sections we know that, for a given number of areas, n, the optimal area sizes $\{\mathrm{A}_1, \mathrm{A}_2, ..., \mathrm{A}_n\}$ would follow the Euler relation in consecutive pairs. Therefore, we could cast this problem as finding the solution of a system of equations which includes the Euler relation between consecutive service region sizes and a market coverage equation (i.e. $\sum_{i=1..n} \mathrm{A}_i = M$).

$$\left(1 - \frac{(\sqrt{2T_i^2 + 4T_i + 4} - 2)}{2T_i}\right) A_i - \left(\frac{\sqrt{2T_i^2 + 4T_i + 4} - 2}{2T_{i+1}}\right) A_{i+1} = 0 \qquad \forall i = 1..n$$

$$\begin{aligned} T_i &= \frac{vA_i}{u + v\sum_{j=1..i} A_j} \qquad \forall i = 1..n \\ \sum_{i=1..n} A_i &= M \end{aligned}$$

One way to solve this system of nonlinear equations is to convert it to a constrained root-finding problem by lifting the market coverage equation as the objective function and leaving the Euler equations as the constraints. We could further transform this constrained root-finding problem to an unconstrained one by implicit substitution of $\mathrm{A}_{i=2,3..n}$. Explicit substitution is impossible due to analytical intractability. We could then solve this root-finding problem numerically in an iterative fashion. The objective function is to find the root of $\digamma(A_1) = \sum\limits_{i=1..n} A_i(A_1) - M = 0$. Here $A_i(A_1)$ represents the implicit substitution. In solving we prefer the Newton's method because of its quadratic local convergence property. Newton's method uses the following iteration formula to calculate the next candidate for the root.

$$A_1^{j+1} = A_1^j - \frac{\digamma(A_1^j)}{\digamma'(A_1^j)} \tag{29}$$

However calculation of $\digamma'(A_1^j)$ is also impossible since an explicit closed form expression of $\digamma(A_1^j)$ is not available. Thus we utilize its discrete approximation which is centered difference approximation of the first-order differential.

$$\digamma'(A_1^j) = \frac{\digamma(A_1^j) - \digamma(A_1^{j-1})}{A_1^j - A_1^{j-1}} \tag{30}$$

With the substitution of (30), the iteration equation in (29) becomes as follows:

$$A_1^{j+1} = A_1^j - \frac{\digamma(A_1^j)\,(A_1^j - A_1^{j-1})}{\digamma(A_1^j) - \digamma(A_1^{j-1})}$$

To calculate the coverage function $\digamma(A_1^j)$, we perform the sequential tiling according to the Euler equations. As a result of this tiling operation, we obtain the terminal boundary, B_{n+1}^j. We could then express $\digamma(A_1^j)$ as follows:

$$\digamma(A_1^j) = B_{n+1}^j - M$$

Accordingly the difference in the coverage function would then be as follows:

$$\mathcal{F}(A_1^j) - \mathcal{F}(A_1^{j-1}) = B_{n^k+1}^j - B_{n^k+1}^{j-1}$$

It is also possible to use other numerical root finding algorithms such as bisection and golden section search methods.

Example 3.1.

Let's now explain this algorithm with an example. In this example we have the demand function defined as $D(x) = 10 + 5x$ and the market region is defined over $[B_1,M]$ =[0,100]. We further choose the fixed cost, i.e. F and f , $F + f = 12,000$. Since we are assuming a linear capacity acquisition function in (17), we could safely exclude it from our analysis since all the solutions will have the same capacity acquisition cost. Initial estimates for the first service region is $\mathrm{A}_1^{j=1}$ $=\mathrm{A}_{\text{min}} = 0$ and $\mathrm{A}_1^{j=2}$ $=\mathrm{A}_{\text{max}} = M/2 = 50$. We also set the optimality and feasibility sensitivity parameters as $\epsilon_{COST} = 2.5 \times 10^{-3}$ and $\epsilon_{BOUND} = 10^{-4}$ Further as suggested in the algorithm above, we start with $n^0 = 3$ service regions and find the optimal and feasible tiling solution for the $n^0 - 1$=2, $n^0 = 3$, and $n^0 + 1 = 4$ service regions as shown in the first iteration ($k = 1$) block in Table 3.1. At the end of first iteration we calculate the n^1 by using the discrete Newton-Raphson approximation using the formulae in the algorithm. So for the first iteration:

$$n^1 = \left\lfloor 3 - \left(\frac{193792.06 - 300353.81}{193792.07 - 2(225928.57) + 300353.81} \right) \right\rceil = \lfloor 5.48 \rceil = 5$$

Note that for each n^k, the optimal value of A_1 is achieved whenever $\mathrm{B}_{n^k+1} = 100$. For faster convergence, we have eliminated the division by two in the discrete approximation formula (28). When the same procedure is repeated, at the end of the second iteration quadratic estimate for the optimal number of service regions is found as 6 for which we know the total cost already. The difference of $TC(n = 6)$ and $TC(n = 5)$ is 386 and thus $\frac{386}{160628.91} < \epsilon_{COST} = 2.5 \times 10^{-3}$; hence, the stopping condition is satisfied and the optimal number of service regions is $n^{k=2} = 6$ with the size of service regions $A_{i=1..6}$= $[31.1, 17.5, 14.7, 13.2, 12.2, 11.4]$

(k)	(n_k)	(j)	A(1)	B(n_k+1)	TC(n_k)	(n_k+1)
1	2	1	0	0		5
		2	50	77.5524		
		3	64.6581	99.7161		
		4	64.65812	100	**300353.81**	
	3	1	0	0		
		2	50	100.7771		
		3	49.61452	100.0139		
		4	49.60752	100	**225928.57**	
	4	1	0	0		
		2	50	121.575		
		3	41.12688	100.4475		
		4	40.93895	99.9999		
		5	40.939	100	**193792.06**	
2	4	1	0	0		6
		2	41.12688	121.575		
		3	41.12688	100.4475		
		4	40.93895	99.9999		
		5	40.939	100	**166221.66**	
	5	1	0	0		
		2	50	140.7317		
		3	35.5287	100.9203		
		4	35.1941	99.9995		
		5	35.1943	100	**160628.94**	
	6	1	0	0		
		2	50	158.6702		
		3	31.5119	101.4013		
		4	31.0595	99.9988		
		5	31.0599	100	**160242.91**	

Table 3.1. Iterations for the constructive solution (Example 3.1).

3.4.2 Improvement Based Solution Approach

Steepest Descent Method for the Allocation Decisions

Rather than building the solution from scratch, one can start from an initial solution and then iteratively improve it via gradient based improvement tech-

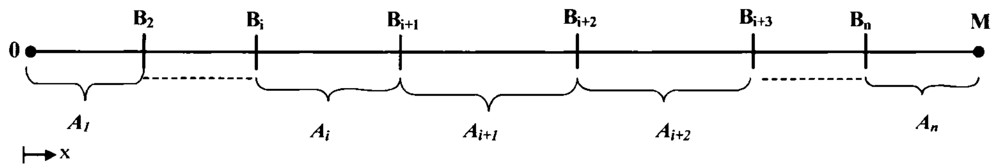

Figure 3.6: A feasible solution for the allocation problem in single dimension.

niques. For this we use the Principle of Optimality implied by the Bellman's equation in (17). More specifically, at any given state (B_i) the optimal decisions ($\mathrm{A}_i, B_{i+1}, \lambda_i$) would be a minimizing solution not only for the total cost in the region $[B_i, B_{i+2}]$, but also for the total cost in the region $[B_i, M]$ if B_{i+2} is optimal (Figure 3.6). Given B_i and B_{i+2}, optimal ($\mathrm{A}_i, B_{i+1}, \lambda_i$) can be found with the first order condition in (26). Optimality of such a minimizing solution is conditional on the optimality of both the B_i and B_{i+2}. Therefore, in order to obtain jointly optimal ($\mathrm{A}_i, B_{i+1}, \lambda_i$), we need to optimize B_i and B_{i+2} as well. This is performed in a recursive manner which could be in forward or backward direction.

Now we provide a formal algorithm of the steepest descent method for the allocation decisions for a given n. Note that to determine the optimal number of service regions, we could use the discrete Newton-Raphson method as in the constructive algorithm presented in the previous section.

Steepest Descent Solution Algorithm - Allocation Decisions (Forward direction)

Step 1. Initialize the model parameters and variables

k : index for the iterations Set $k = 1$

ϵ_{COST} : epsilon parameter for optimality stopping decisions

$TC(k = 0) = \infty$

Step 2. Start from an initial solution $(B_i^1, A_i^1, i = 1, 2, ..n)$

Do While $(|TC(k) - TC(k-1)| \geq \epsilon_{COST})$:

$k = k + 1$

For $i = 1$ to $n - 1$, REPEAT

· Solve $\frac{\partial TC(A_i', B_i)}{\partial A_i'} + \frac{dTC(B_{i+2} - A_i' - B_i, B_i + A_i')}{dA_i'} = 0$ for A_i'

· Set $A_i^k = A_i'$, $B_{i+1}^k = B_i^k + A_i'$,and $A_{i+1}^k = B_{i+2}^k - A_i' - B_i^k$

Return

Step 3. Terminate with optimal solution $\left(B_i^k, A_i^k, i = 1, 2, ..n\right)$

In the above algorithm, the for-loop considers one pair of service regions ($[B_i^k, B_{i+1}^k]$ and $[B_{i+1}^k, B_{i+2}^k]$) at a time and repeats $n - 1$ times (i.e. total number of pairs for n service regions) the steps in the loop. The differential equation in the **Step 2.,** called Euler step**,** determines such an A_i', which optimally divides the region $[B_i^k, B_{i+2}^k]$ into two service regions, namely any other allocation would result in a higher total cost.

Example 3.2.

We now illustrate this improvement based solution methodology with an example. In this example, we have three service regions which are initially sized as A_1=30, A_2=30, and A_3=40 to cover a market region defined over $[B_1, M]$ =[0,100] as shown in Figure 3.7. We also set our optimal solution sensitivity parameter as $\epsilon_{COST} = 1.0$. Since there are three service regions, the algorithm stipulates that the Euler step in Step 2. is to be repeated twice. Thus there are two steps $j = 1, 2$ for each iteration k.

The demand density function defined over the market is $D(x) = 10 + 5x$. Let's now perform the improvement iterations starting from left boundary

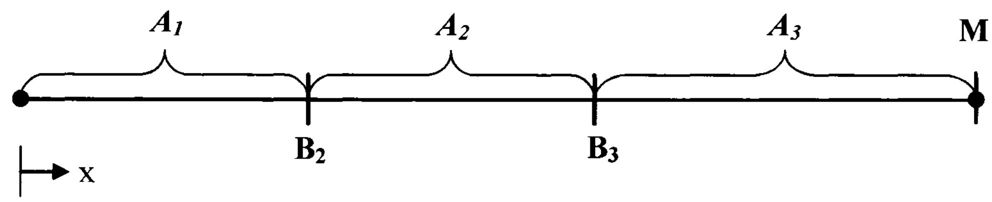

Figure 3.7: An illustration of districting a market region defined over $[B_1 = 0, M = 100]$ with three service regions.

($B_1 = 0$) onwards (i.e. in the forward direction).

We first consider $[B_1, B_3] = [0, 60]$. Solving the Euler equation in (26) for A_1, leads to A_1=38.54 and $B_2 = 38.54$. This solution is conditional on the B_3 thus we move one area forward and consider $[B_2, M] = [38.54, 100]$. Again via (26), optimal solution for second service region can be found as A_2=34.26 ($B_3 = 72.8 = B_2$+34.26). Hence this iteration results in: A_1=38.54 , A_2=34.26, and A_3=27.2. After repeating the same steps in iteration 1, we obtain $A_1 =$ 46.90, $A_2 =$ 28.99, and $A_3 =$ 24.11. If we continue with the same steps, we converge to optimal solution of $A_1 = 49.60$,$A_2 = 27.35$, and $A_2 = 23.05$ in the sixth iteration after verifying the optimality condition. These iteration steps are shown in Table 3.2. Note that $|TC(k = 6) - TC(k = 5)| < 1 = \epsilon_{COST}$, hence the solution at the end of sixth iteration is optimal. The difference in the total cost of the final solution in this example differs from the previous one in the constructive approach example where $n = 3$. This difference arises because in this example we excluded the fixed costs, i.e. $3(F + f) = 36,000$, from the objective function.

Iteration No (k)	Euler Equation Iteration(i)	A(I)	A(I+1)	B(I)	B(I+2)	A'(I)	TC(k)
1	1	30.00	30.00	0.00	60.00	38.54	
	2	21.46	40.00	38.54	100.00	34.26	196291.1
2	1	38.54	34.26	0.00	72.80	46.90	
	2	25.90	27.20	46.90	100.00	28.99	187546.2
3	1	46.90	28.99	0.00	75.89	48.91	
	2	26.98	24.11	48.91	100.00	27.76	186969.4
4	1	48.91	27.76	0.00	76.68	49.43	
	2	27.25	23.32	49.43	100.00	27.45	186930.8
5	1	49.43	27.45	0.00	76.88	49.56	
	2	27.32	23.12	49.56	100.00	27.37	186928.2
6	1	49.56	27.37	0.00	76.93	49.60	
	2	27.34	23.07	49.60	100.00	27.35	186928.1

Table 3.2. Iteration results for improvement based algorithm for the service region sizes (Example 3.2).

In the above algorithm we improve the allocation decisions one at a time. Another alternative is to determine a descent direction for all of the allocation decisions and then decide on the joint step size. Since our problem is constrained, we need to identify the descent directions which are feasible. These directions are

$$d_{A_i} = -\left(\frac{\partial TC(A_i, B_i)}{\partial A_i} + \frac{dTC(B_i + A_i, B_{i+2} - A_i - B_i)}{dA_i}\right) \text{ for } i = 1..n-1$$

In order to see that these directions are feasible and steepest descent directions, we first write the Lagrangian function of the $P1$.

$$L(B, A, \lambda, \mu, \nu) = \sum_{i=1}^{n} TC_i(B_i, A_i) + \sum_{i=1}^{n} \lambda_i(B_{i+1} - B_i - A_i) - \sum_{i=1}^{n-1} \mu_i(B_{i+1}) - \sum_{i=1}^{n} \nu_i A_i$$

where $(\lambda, \mu \geq 0, \nu \geq 0)$ are the Lagrange multipliers associated with the state equations and non-negativity constraints. From the KKT conditions we know that for B_{i+1},$A_i > 0$, the associated lagrange multipliers must be zero, i.e. $\mu_i = \nu_i = 0$. Thus if the solution of the $P1$ is an internal solution with respect to the solution set implied by **non-negativity constraints**, then we could ignore the multipliers (μ, ν). In this case, the gradient of the Lagrangian function could be expressed as follows:

$$
\begin{aligned}
\nabla L &= (\nabla_{A_{i=1...n}} L, \nabla_{B_{i=2...n}} L, \nabla_{\lambda_{i=1...n}} L) && (31) \\
& \quad \text{where} \\
\nabla_{A_i} L &= \frac{\partial TC_i(B_i, A_i)}{\partial A_i} - \lambda_i && i = 1...n \quad (32) \\
\nabla_{B_i} L &= \frac{\partial TC_i(B_i, A_i)}{\partial B_i} + \lambda_{i-1} - \lambda_i && i = 2...n \quad (33) \\
\nabla_{\lambda_i} L &= B_{i+1} - B_i - A_i && i = 1...n \quad (34)
\end{aligned}
$$

We know that if (B_i, A_i, λ_i) is an optimal solution to $P1$, then $\nabla L = 0$. Thus we first perform the following assignments, to obtain $\nabla_{\lambda_i} L = 0$.

$$
\begin{aligned}
B_{i+1} &:= B_i + A_i && i = 1...n-1 \quad (35) \\
A_n &:= B_{n+1} - B_n && (36)
\end{aligned}
$$

This assignments will ensure the feasibility of our descent direction. Since we have determined B_{i+1} for $i = 1..n-1$ in (35) for directional feasibility, they are no longer independent variables. We could now solve (33) for $\nabla_{B_i} L = 0$. This brings us following relations (note the change of indices).

$$\begin{aligned} \nabla_{B_{i=2..n}} L &= 0 \\ \lambda_i &= -\frac{\partial TC_i(B_i, A_i)}{\partial B_i} + \lambda_{i+1} \qquad i = 1...n-1 \end{aligned} \tag{37}$$

From the minimum principle, $\nabla_{A_i} H(B_i, A_i, \lambda_i) = 0$, we have the following relations.

$$\lambda_{i+1} = \frac{\partial TC_{i+1}(B_{i+1}, A_{i+1})}{\partial A_{i+1}} \qquad i = 1...n-1 \tag{38}$$

When we substitute expression for λ_{i+1} in (38) to (37), and then λ_i into (32), we obtain the following gradients.

$$\nabla_{A_i} L = \frac{\partial TC_i(B_i, A_i)}{\partial A_i} - \frac{\partial TC_{i+1}(B_{i+1}, A_{i+1})}{\partial A_{i+1}} + \frac{\partial TC_{i+1}(B_{i+1}, A_{i+1})}{\partial B_{i+1}} \; i = 1...n-1 \tag{39}$$

From the imposed feasibility conditions in (35), we can re-express (39) as follows.

$$\nabla_{A_i} L = \frac{\partial TC_i(B_i, A_i)}{\partial A_i} + \frac{dTC_{i+1}(B_i + A_i, B_{i+2} - B_i - A_i)}{dA_i} \qquad i = 1...n-1$$

where

$$\begin{aligned} \frac{dTC_{i+1}(B_i + A_i, B_{i+2} - B_i - A_i)}{dA_i} &= \frac{\partial TC_{i+1}(B_{i+1}, A_{i+1})}{\partial A_{i+1}} \frac{\partial A_{i+1}}{\partial A_i} \\ &\quad + \frac{\partial TC_{i+1}(B_{i+1}, A_{i+1})}{\partial B_{i+1}} \frac{\partial B_{i+1}}{\partial A_i} \end{aligned}$$

Note that we do not iterate A_n since it is determined by (36). Thus the gradient of the TC at a feasible and non-zero solution could be expressed as follows.

$$\nabla_{A_i} TC = \frac{\partial TC_i(B_i, A_i)}{\partial A_i} + \frac{dTC_{i+1}(B_i + A_i, B_{i+2} - B_i - A_i)}{dA_i} \quad i = 1...n-1 \quad (40)$$

Using this feasible descent direction, we could set up the following algorithm.

Steepest Descent Solution Algorithm - All Allocation Decisions

Step 1. Initialize the model parameters and variables

k : index for the iterations Set $k = 1$

ϵ_{COST} : epsilon parameter for optimality stopping decisions

$TC(k = 0) = \infty$

Step 2. Start from an initial solution $(B_i^1, A_i^1, i = 1, 2, ..n)$

Do While $(|TC(k) - TC(k-1)| \geq \epsilon_{COST})$:

$k = k + 1$

Gradient Calculation

Calculate $d_{A_i}^k = -\left(\frac{\partial TC(B_i^k, A_i^k)}{\partial A_i^k} + \frac{dTC(B_i^k + A_i^k, B_{i+2}^k - A_i^k - B_i^k)}{dA_i^k}\right)$ for $i = 1..n-1$

Parametric tiling

For $i = 1$ to $n - 1$, REPEAT

$A_i' = A_i^k + \lambda^k d_{A_i}^k$, $B_{i+1}' = B_i' + A_i'$

Return

Line Search

Solve $\min\limits_{\lambda^k} TC = \sum\limits_{i=1}^{n} TC_i(B_i', A_i')$ using a line search method

Finalize tiling

For $i = 1$ to $n-1$, REPEAT

$$A_i^{k+1} = A_i^k + \left(\lambda^k\right)^* d_{A_i}^k,\ B_{i+1}^{k+1} = B_i^{k+1} + A_i^{k+1}$$

Return

Step 3. Terminate with solution $\left(B_i^k, A_i^k, i = 1, 2, ..n\right)$

The results are displayed in Table 3.3. Since we are iterating all the allocation decisions at the same time, number of iterations ($k = 9$) is more than the separate iteration. However, here we are only making 9 line searches rather than the 12(=2×6) in individual iteration approach. In terms of convergence, results seem to be consistent.

Iteration No (k)	A1(k)	A2(k)	d_{A1}(k)	d_{A2}(k)	λ*(k)	TC(k)
1	30.00	30.00	1281.14	3019.21	0.0030	228,917.96
2	33.90	39.20	2240.85	-667.58	0.0049	205,320.31
3	44.85	35.94	1575.13	-2243.50	0.0039	195,055.85
4	50.93	27.28	-120.47	-283.90	0.0025	187,184.27
5	50.63	26.57	-204.43	60.90	0.0037	187,024.38
6	49.87	26.80	-104.78	149.25	0.0037	186,962.96
7	49.48	27.35	11.42	26.92	0.0025	186,930.40
8	49.51	27.42	19.42	-5.79	0.0038	186,928.94
9	49.58	27.40	10.14	-14.44	0.0037	186,928.38

Table 3.3. Iteration results for improvement based algorithm for the service region sizes with joint iteration (Example 3.3).

3.5 Conclusions

In this chapter we developed a dynamic programing formulation for the fixed-charge continuous location-allocation model on a line. Based on this formulation, we have characterized the optimal solution using the *minimum principle* of the optimal control theory. These optimality conditions evidenced that when the capacity acquisition cost is linear with the output volume, the central planner's allocation decisions concur with the customers' patronizing decisions.

From this result, we have designed two alternative solution methodologies for the problem on a line. First method is the single-step shooting algorithm which is frequently used in the solution of two-point boundary value problems. This method is efficient when there are large number of service regions. However, this method is restricted to solving these problems in the single dimension. Second solution method is the improvement based steepest-descent solution approach based on the allocation decision iterations. In this approach, we start with an initial solution and then iteratively improve this solution until an optimal allocation solution is obtained. When compared with the steepest-descent algorithm, the constructive solution method is powerful in its convergence rate. However, as described in the next two chapters as well as in Chapter 7, constructive solution method is not easily extendable to planar n-facility problems. However, through the introduction of new allocation decision definitions, we will demonstrate its extension to planar 2-facility cases for Euclidean and Manhattan-metric cases.

Chapter 4

Planar Model: 2-Facility and Euclidean-Metric Case

4.1 Introduction

The planar location-allocation problem differs from its counterpart in the single-dimension with two main aspects. Firstly, the representation of the allocation decisions, which are point representations in the single-dimensional setting, becomes point-set representation in the planar setting. An example illustration of the 2-facility planar location-allocation problem solution is provided in Figure 4.1. The straight line separating the $\mathcal{A}_1$ and $\mathcal{A}_2$ is a point-set as compared to single point (i.e. boundary points) in the single dimensional setting. Second difference is related to the improvement of allocation decisions. Since the service regions are separated by boundary points in the single dimensional setting, any change in the allocation decisions is equivalent to iterating these points. However, in the planar setting we iterate a straight line, which requires a different approach. Due to these differences, both the improvement and constructive solution procedures for the planar 2-facility case are different than their single-dimensional counterparts.

The purpose of this chapter is two fold. First, we establish the differences between the single-dimensional and planar location-allocation problems with a special emphasis on the representation of allocation decisions. This step paves the way for designing constructive and improvement solution procedures for the planar 2-facility location-allocation problem in the allocation variable space. Second objective of this chapter is to develop two main classes of solu-

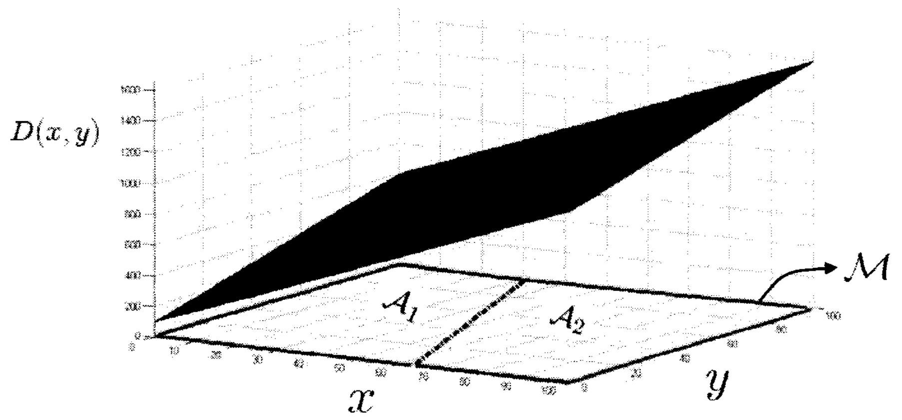

Figure 4.1: An illustration of the solution to the planar location-allocation problem with two service regions

tion procedures for the 2-facility case. One class is the constructive solution approach and the other class is the improvement based solution approaches. Whereas the constructive solution approach constitutes an extension to the one introduced for single dimensional problems, improvement solution approaches are significantly different and constitutes a starting point for developing methods for more general cases, namely n-facility planar location-allocation problems.

In what follows, Section 4.2. presents the notation to be used in this chapter as well as the representation of the allocation decisions for two distance measures, namely Euclidean-metric and Squared Euclidean-metric. Section 4.3. presents the traditional modeling approach in the location variable space and our *alternative* modeling in the allocation variable space. This section also presents analytical properties of the solutions for each modeling approach. Section 4.4. presents two categories of solution methods. First category is the constructive solution method, namely the *shooting algorithm*. This shooting algorithm is an adaptation of the single-dimensional version from the previous chapter for two different distance measures. Euclidean-metric and Squared Euclidean-metric cases share the similar constructive solution algorithm, due to the identical form of their allocation decisions, i.e., straight line. They, however, differ only in the their single-facility location step. Second category includes two improvement based solution approaches, namely the steepest-descent algorithm and modified Newton's method.

Throughout this chapter, we consistently use the demand density function shown in Figure 4.1, which is a linear demand density function. Note that this is not a limitation of the results in this chapter, rather it serves as a simplification in the presentation of the results and interpretation of the solutions of the algorithms. Accordingly, all the results developed here are valid for any *Lipschitz continuous* demand density function over the market region $\mathcal{M}$.

4.2 Description of Parameters and Notation

In what follows, we first describe the parameters and decision variables of the planar location-allocation problems and briefly discuss the differences from their single dimensional counterparts. Note that most of the following discussion is also applicable to more than 2-facility cases. Some of these definitions can be observed in Figure 4.1, where a representative solution to the planar location-allocation problem for two service regions is plotted with a linear demand density function.

This section introduces the following notation and parameter definitions:

Parameters:

$\mathbf{x}$: a point in the two dimensional space $\mathbf{x} \equiv (x, y)$

$\mathcal{M}$: Two dimensional market area (assumed to be a closed and compact set)

$D(\mathbf{x})$: Demand density function over the two-dimensional market region $\mathcal{M}$ ($D(\mathbf{x}) \equiv D(x, y)$)

$d_p(\mathbf{x}_1, \mathbf{x})$: Shortest distance between $\mathbf{x}_1$ and $\mathbf{x}$ for a given distance measure p ($\mathbf{p} = \mathbf{L}_2$ denotes *Euclidian-metric*, $\mathbf{p} = \mathbf{L}_2^2$ denotes *Squared Euclidian-metric*)

In two-dimensional formulations, we have two main decision variables: *Location* decisions and *Allocation* decisions. These decision variables are defined below.

Decision Variables:

$\mathbf{x}_1, \mathbf{x}_2$: Locational coordinates of the facilities in service region 1 and 2, i.e. $\mathbf{x}_1 \equiv (x_1, y_1)$, $\mathbf{x}_2 \equiv (x_2, y_2)$

$\mathcal{A}_1, \mathcal{A}_2$: Service regions 1 and 2 (assumed to be closed sets)

$\mathbf{x}_1^*, \mathbf{x}_2^*$: Optimal locations given the allocation decisions $\mathcal{A}_1$ and $\mathcal{A}_2$

Given location decisions $(\mathbf{x}_1, \mathbf{x}_2)$, optimal allocation decisions can be found using the *nearest-neighbor property.* In this case, they would be the point sets defined as follows.

$$\begin{aligned}
\mathcal{A}_1 &= \{\mathbf{x} \mid d_p(\mathbf{x}_1, \mathbf{x}) \leq d_p(\mathbf{x}_2, \mathbf{x}),\ \mathbf{x} \in \mathcal{M}\} \\
\mathcal{A}_2 &= \{\mathbf{x} \mid d_p(\mathbf{x}_2, \mathbf{x}) \leq d_p(\mathbf{x}_1, \mathbf{x}),\ \mathbf{x} \in \mathcal{M}\}
\end{aligned}$$

Traditionally, continuous location-allocation problems have been modeled in the *location variable space.* Accordingly allocation decisions are optimally decided given the location variables, such as the case of nearest-neighbor assignment. In order to solve the problem in the allocation space, however, we need to define *additional constructs* to represent these allocation decisions. First step is to define the **Allocation Line (BR)** as follows:

BR : intersection point set of the allocation decisions $\mathcal{A}_1$ and $\mathcal{A}_2$ (i.e. $BR = \mathcal{A}_1 \cap \mathcal{A}_2$). Herein it will be referred as the *Allocation Line.* For the nearest-neighbor solution case, this allocation line could be expressed as below.

$$BR = \{\mathbf{x} \mid d_p(\mathbf{x}_1, \mathbf{x}) = d_p(\mathbf{x}_2, \mathbf{x}),\ \mathbf{x} \in \mathcal{M}\} \qquad \text{for } p = L_2 \text{ and } p = L_2^2$$

Since we are assuming that service regions are closed sets, all points on the BR must be connected. It can be shown that nearest-neighbor solution of BR is a straight line for both the Euclidian-metric distance measure (i.e. $p = L_2$) and the Squared Euclidian-metric distance measure (i.e. $p = L_2^2$).[3] From now on, we will adopt this form of the allocation line, i.e. straight line, for Euclidean-metric based distance measures and introduce the following parameters and formulations for BR.

BR can be characterized by using the *slope* and *intercept* parameters.

$$BR = \{\mathbf{x} | \mathbf{x} = (x, y) \in \mathcal{M} \text{ and } y = ax + b\}$$

where

a : slope of the straight line BR

b : intercept of the straight line BR

[3]For other metrics, BR is a different shape than a straight line, which will be discussed for the Manhattan-metric case in Chapter 7.

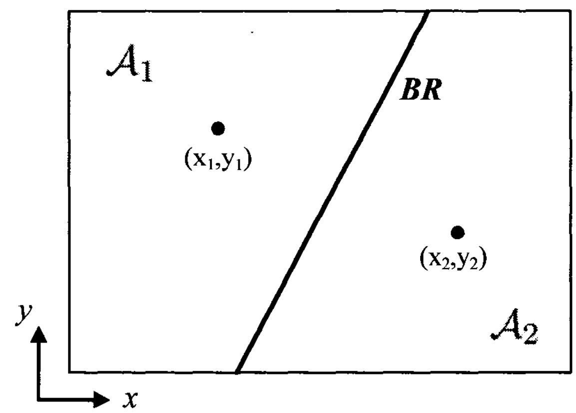

Figure 4.2: Separating allocation set BR (a.k.a. Allocation Line) for Euclidean-metric based distance measures.

Using the above notation, we define the **functional form** of BR as $br(x)$.

$$br(x) = y = ax + b \qquad \text{for } \forall(x, y) \in BR$$

Accordingly the *inverse function* of $br(\cdot)$, namely $br^{-1}(\cdot)$, is as follows:

$$br^{-1}(y) = x = \frac{y - b}{a} \qquad \text{for } \forall(x, y) \in BR$$

Above formulations allow us to represent *the allocation line mapping in closed form.* Next, we will define specific sets on the $x-$ and $y - axis$ to define the **domains of these mappings.**

Thus far we have not assumed any particular shape for the market region $\mathcal{M}$, but, without loss of generality, we hereafter consider a *square-shaped market region.* While this may seem a rather stringent assumption, all of the subsequent sections' results can be shown to be adapted easily, provided that $\mathcal{M}$ is fully known in its shape and size. With this assumption, we now segment the $x-$ and $y-axis$ into sections to differentiate point sets which form the domain of the functional form of BR from those which do not.

Any point $\mathbf{x} \in BR$ could be represented by its coordinates x and y from the origin. Let's define following sets associated with BR.

$$
\begin{aligned}
X_{BR} &= \{x \mid \mathbf{x} = (x, y) \in BR\} \\
X_{A1} &= \{x \mid \mathbf{x} = (x, y) \in \mathcal{A}_1 \text{ and } \mathbf{x} = (x, y) \notin BR\} \\
X_{A2} &= \{x \mid \mathbf{x} = (x, y) \in \mathcal{A}_2 \text{ and } \mathbf{x} = (x, y) \notin BR\}
\end{aligned}
$$

$$
\begin{aligned}
Y_{BR} &= \{y \mid \mathbf{x} = (x, y) \in BR\} \\
Y_{A1} &= \{y \mid \mathbf{x} = (x, y) \in \mathcal{A}_1 \text{ and } \mathbf{x} = (x, y) \notin BR\} \\
Y_{A2} &= \{y \mid \mathbf{x} = (x, y) \in \mathcal{A}_2 \text{ and } \mathbf{x} = (x, y) \notin BR\}
\end{aligned}
$$

These sets are illustrated in the Figure 4.3 for the case when BR is a straight line (i.e. cases L_2 and L_2^2).

Further, we define four special points, namely **extreme points of the domain of** BR $(X_{\min}, X_{\max}, Y_{\min}, Y_{\max})$, for use in the subsequent sections of this chapter. These points are illustrated in Figure 4.3.

$$
\begin{aligned}
[X_{\min}, X_{\max}] &= X_{BR} = \{x \mid \mathbf{x} = (x, y) \in BR\} \\
[Y_{\min}, Y_{\max}] &= Y_{BR} = \{y \mid \mathbf{x} = (x, y) \in BR\}
\end{aligned}
$$

Lastly, we define the following functions using **functional form of the allo-**

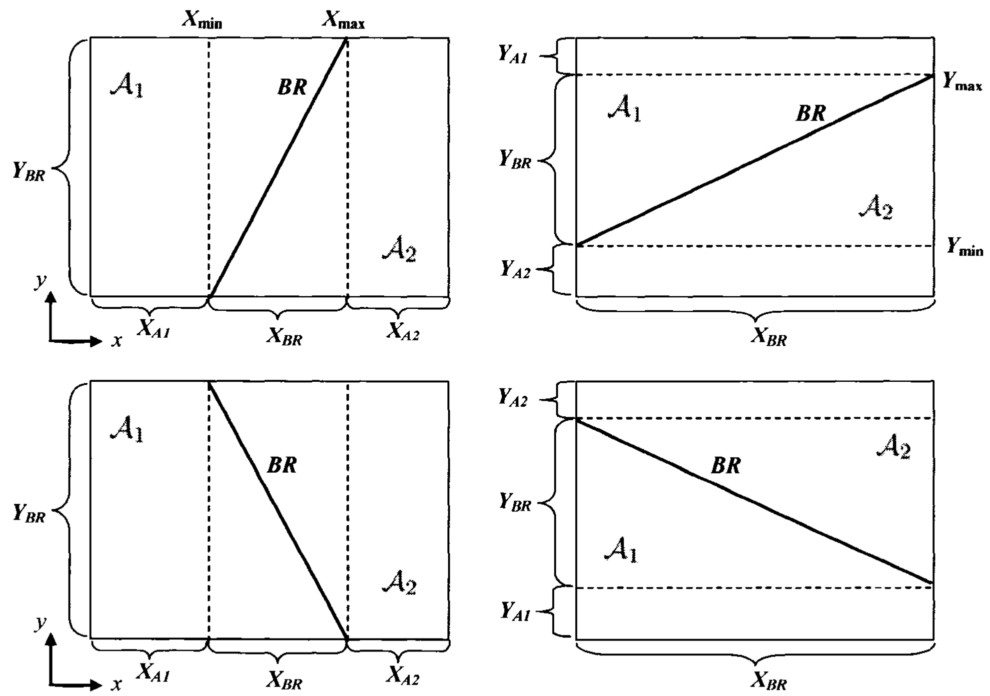

Figure 4.3: Sets specifying the domain of functions of the allocation line (BR).

cation line (BR) and above sets as corresponding domains.

$$BR = (A^x(y), y) = (x, A^y(x)) \qquad \text{for } x \in X_{BR} \text{ and } y \in Y_{BR}{}^4$$

These functions are illustrated in the Figure 4.4. Hereon, we will be referring to these functions (namely $A^y(x)$ and $A^x(y)$) as the **Single-dimensional Allocation Decisions**. Although, $A^x(y)$ and $A^y(x)$, represents the same line, BR, we will use both of them in our model formulation to handle the degenerate cases of BR where $A^x(y)$ or $A^y(x)$ is not defined. These degenerate cases are illustrated in Figure 4.5.

[4]To better understand this equality, consider a point $x = x_0$ and $y = y_0$ on BR. Then, from the definitions of $A^x(y)$ and $A^y(x)$ in Figure 4.4, this equality holds true.

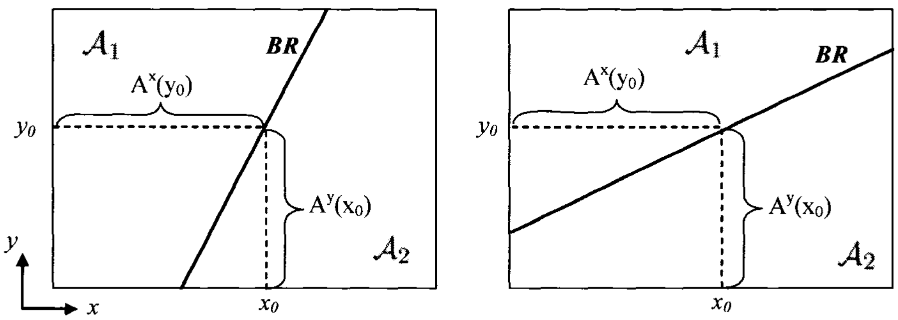

Figure 4.4: Illustration of the single-dimensional allocation decisions $A^y(x)$ and $A^x(y)$ for two cases of X_{BR} and Y_{BR}.

4.3 Alternative Modeling Approaches for 2-Facility Case

In this section we will describe two alternative modeling approaches for planar location-allocation problems; these approaches are also applicable to cases with more than two facilities. In the location-allocation problem, there are two *independent* sets of decisions: *Locations* of the facilities and *Allocations* of demand to these facilities. Despite the innate independence of these decisions, location-allocation problems are traditionally modeled in the location variable space and allocation decisions are decided optimally given these locations. One exception, to the best of our knowledge, is the Sherali and Tuncbilek (1992) which models the location-allocation problem with discrete demand in the allocation variable space for the Squared-Euclidean metric (L_2^2) case.

In order to juxtapose the two modeling approaches, **Location Variable Space (LVS)** and **Allocation Variable Space (AVS)**, we first present the generic model formulation in the joint variable space. In all of these models, we use $(\mathbf{x}_1,\mathbf{x}_2)$ to denote the location decisions and, $A^y(x)$ and $A^x(y)$ to denote the allocation decisions. For sake of notational simplicity, we use implicit representation of the objective function in this section. In the following sections, individual terms of the objective function will be specified in more detail.

The generic model in joint variable space is as follows.

Location-Allocation Model (LAM):

$$\min_{\substack{A^x(y),A^y(x)\\ \mathbf{x}_1=(x_1,y_1),\mathbf{x}_2=(x_2,y_2)}} \quad TC(A^y(x), A^x(y), \mathbf{x}_1, \mathbf{x}_2)$$

subject to

$$(A^x(y), y) = (x, A^y(x)) \qquad for\ x \in X_{BR},\ y \in Y_{BR}$$

where

$TC(A^y(x), A^x(y), \mathbf{x}_1, \mathbf{x}_2)$: is the total cost function defined over the (2×1) column vectors, $\mathbf{x}_1$ and $\mathbf{x}_2$, and functionals, $A^y(x)$ and $A^x(y)$, which are defined in the preceding section.

Since the feasible region is composed of affine relations, LAM is a biconvex programming problem; when we fix the allocation decisions, it becomes a multifacility planar location problem and fixing location decisions transforms it into a transportation problem. It can be further shown that LAM is non-convex problem (for L_2 and L_2^2 as well as the Manhattan-metric-L_1). To illustrate this consider the solution for a square-shaped market region $\mathcal{M}$ in the Figure 4.5 where $D(x,y) = constant$, i.e. uniform demand density function. These solutions (left and right) are both minimizers of TC and existence of other feasible solutions which are worse in terms of TC can be shown.

Next section analyzes the model in the location variable space, where the allocation decisions are optimized given the location decisions.

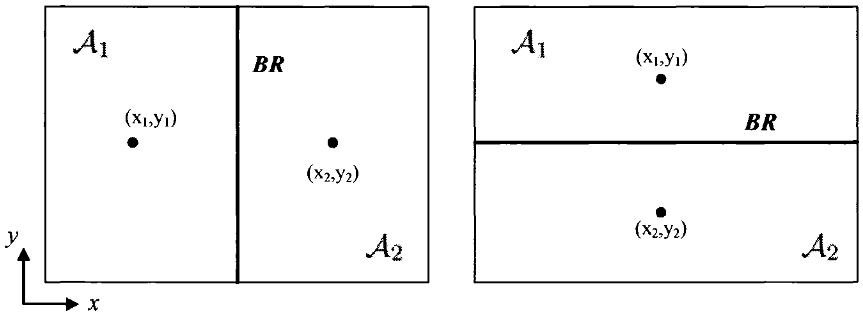

Figure 4.5: Locally optimal solutions when $D(x, y)$ is a uniform demand density function.

4.3.1 Modeling in Location Variable Space

Location prioritized model is based on transforming LAM into an equivalent form with decisions variables as the location decision *only*. Therefore following problem is equivalent to the LAM:

$$\min_{\mathbf{x}_1,\mathbf{x}_2} \widetilde{TC}(\mathbf{x}_1, \mathbf{x}_2)$$

where

$$\widetilde{TC}(\mathbf{x}_1, \mathbf{x}_2) = \min_{A^x(y),A^y(x)} \left\{ \begin{array}{c} TC(A^y(x), A^x(y), \mathbf{x}_1, \mathbf{x}_2)| \\ (A^x(y), y){=}(x, A^y(x)) \text{ for } x \in X_{BR},\ y \in Y_{BR} \end{array} \right\}$$

This is equivalent to optimizing first over the allocation decisions and then over the location decisions. Ideally, we would solve $TC(A^y(x), A^x(y), \mathbf{x}_1, \mathbf{x}_2)$ only for the allocation decisions and substitute optimal allocation decisions to obtain $\widetilde{TC}(\mathbf{x}_1, \mathbf{x}_2)$. In the single-dimensional case, optimal allocation decision, which is a single boundary point, can be expressed as $(\frac{x_1+x_2}{2})$ which leads to a closed form of $\widetilde{TC}(\mathbf{x}_1, \mathbf{x}_2)$. However, a closed form expression of $\widetilde{TC}(\mathbf{x}_1, \mathbf{x}_2)$ in two-dimensional setting is difficult to obtain, besides being unnecessary.

Instead, we include a condition that characterizes the optimal allocation decisions in the constraint set. Given locations $\mathbf{x}_1$ and $\mathbf{x}_2$, optimal allocation decisions would satisfy the following condition. This condition stipulates that each point on the allocation line is **equidistant** from the locations.

$$d_p(\mathbf{x}_1, (x, A^y(x))) = d_p(\mathbf{x}_2, (A^x(y), y)) \qquad \text{for } x \in X_{BR} \text{ and } y \in Y_{BR}$$

We now present the *location-allocation problem in the location variable space.*

LAM- Location Variable Space (LAM-LVS):

$$\min_{\mathbf{x}_1=(x_1,y_1),\mathbf{x}_2=(x_2,y_2)} TC(A^y(x), A^x(y), \mathbf{x}_1, \mathbf{x}_2)$$

$$\begin{aligned} s.t.& \\ (A^x(y), y) &= (x, A^y(x)) \qquad \text{for } x \in X_{BR} \text{ and } y \in Y_{BR} \\ d_p(\mathbf{x}_1, (x, A^y(x))) &= d_p(\mathbf{x}_2, (A^x(y), y)). \qquad \text{for } x \in X_{BR} \text{ and } y \in Y_{BR} \quad (41) \end{aligned}$$

In addition to the variable space, a notable difference between LAM and LAM-LVS is the last constraint (41). This constraint conditions the optimality of allocation decisions on the location decisions, while making the allocation decisions endogenous decision variables and leaving the location variables as exogenous decision variables. Furthermore, LAM-LVS would be a convex programming problem if constraint (41) had been affine. Since LAM-LVS and LAM are equivalent problems, LAM-LVS is also a nonconvex problem. We now characterize the first order necessary conditions for LAM-LVS (assuming the presence of only the transportation costs in the objective) in the case of

L_2 and L_2^2.

Proposition 4.1.

The optimal locations of the two facilities ($\mathbf{x}_1^*$ and $\mathbf{x}_2^*$)*, given the allocation decisions, satisfy the following conditions when the distance measure is based on the* **Squared Euclidean-Metric** ($\mathbf{L}_2^2$) :

$$(x_i^*, y_i^*) = \left(x_i^G, y_i^G\right) \qquad \text{for } i = 1, 2$$

where x_i^G *and* y_i^G *are the* $x-$ *and* $y-$ *dimensional centroids of* $A_{i=1,2}$ *with respect to* $D(x)$.

$$x_i^G = \frac{\int_{\mathcal{A}_i} x D(\mathbf{x}) d\mathbf{x}}{\int_{\mathcal{A}_i} D(\mathbf{x}) d\mathbf{x}} \qquad \text{and} \qquad y_i^G = \frac{\int_{\mathcal{A}_i} y D(\mathbf{x}) d\mathbf{x}}{\int_{\mathcal{A}_i} D(\mathbf{x}) d\mathbf{x}} \qquad \text{for } i = 1, 2$$

Proof.

Proof can be found in Appendix 4. ■

Proposition 4.2.

The optimal locations of the two facilities ($\mathbf{x}_1^*$ and $\mathbf{x}_2^*$)*, given the allocation decisions, satisfy the following conditions when the distance measure is based on the* **Euclidean-Metric** ($\mathbf{L}_2$) :

$$\int_{\mathcal{A}_i} \frac{(x_i^* - x)}{||\mathbf{x}_i^* - \mathbf{x}||} D(\mathbf{x}) d\mathbf{x} = 0 \qquad \text{for } i = 1, 2$$

$$\int_{\mathcal{A}_i} \frac{(y_i^* - y)}{||\mathbf{x}_i^* - \mathbf{x}||} D(\mathbf{x}) d\mathbf{x} = 0 \qquad \text{for } i = 1, 2$$

Proof.

Proof can be found in Appendix 4. ■

4.3.2 Modeling in Allocation Variable Space

Similar to the location-allocation model in the location variable space, the model in allocation variable space is also based on transforming LAM into an equivalent form with decision variables as the allocation decisions *only*. Hence, the following problem is equivalent to the LAM.

$$\min_{A^x(y), A^y(x)} \widetilde{TC}(A^x(y), A^y(x))$$

where

$$\widetilde{TC}(A^x(y), A^y(x)) = \min_{\mathbf{x}_1, \mathbf{x}_2} \{TC(A^y(x), A^x(y), \mathbf{x}_1, \mathbf{x}_2)\} \tag{42}$$

This is equivalent to optimizing first over the location decisions and then over the allocation decisions. Optimal solution to (42) can be expressed in closed form for the single-dimensional case, but the same is not true for the two-dimensional case. In particular, these optimal location solutions satisfy the first order necessary conditions of the LAM-LVS in the previous section (i.e. outlined in the Propositions 4.1. and 4.2.). When these necessary conditions are included in the constraint set of the LAM, we obtain the following *location-allocation model in the allocation variable space.*

LAM- Allocation Variable Space (LAM-AVS):

$$\min_{A^x(y),A^y(x)} TC(A^y(x), A^x(y), \mathbf{x}_1^*, \mathbf{x}_2^*)$$

$$
\begin{aligned}
& s.t. \\
(A^x(y), y) \;&=\; (x, A^y(x)) \qquad \text{for } x \in X_{BR} \text{ and } y \in Y_{BR} \\
\mathbf{x}_i^* \;&=\; \arg\min_{(\mathbf{x}_i)} \int_{\mathcal{A}_i} d_p(\mathbf{x}_i, \mathbf{x}) D(\mathbf{x}) d\mathbf{x} \qquad \text{for } i = 1, 2 \qquad (43)
\end{aligned}
$$

In addition to the variable space, a notable difference between LAM and LAM-AVS is the last constraint (43). This constraint conditions the optimality of location decisions on the allocation decisions, while making the location decisions as endogenous decision variables and leaving the allocation variables as exogenous decision variables. LAM-AVS would have been a convex programming problem if constraint (43) had been affine. Since LAM-AVS and LAM are equivalent problems, LAM-AVS is also a nonconvex problem. We now characterize first order necessary conditions for LAM-AVS (assuming the presence of only the transportation costs in the objective). These conditions are in the same form for both Euclidean-metric based distance measures, i.e. L_2 and L_2^2.

First Order Necessary Conditions for LAM-AVS (for $p = L_2$, and L_2^2) :

$$d_p(\mathbf{x}_1^*, (x, A^y(x))) - d_p(\mathbf{x}_2^*, (A^x(y), y)) = 0 \qquad (44)$$

for $\forall x \in X_{BR}$, $\forall y \in Y_{BR}$, and $(A^x(y), y) = (x, A^y(x))$

Note that first order condition of the LAM-AVS is part of the constraint set of LAM-AVS and vice-versa.

4.4 Solution Methodologies

In this section, we describe three alternative solution methods to determine the local optimum solutions for the LAM-AVS. First of these methods is the **Constructive Solution approach** which is analogous to the one presented in the previous chapter for single-dimensional case. The later two approaches are improvement based approaches: **Steepest-Descent method**, and **Modified-Newton method.**[5]

4.4.1 Constructive- Shooting Algorithm

Along the lines of the single-dimensional case in the previous chapter, planar location-allocation problem in the allocation variable space (LAM-AVS) for 2-facility case can be cast as an equivalent system of differential equations, where the unknowns are the single dimensional allocation decisions, namely $A^x(y)$ and $A^y(x)$. Accordingly, this system of differential equations can be solved using the shooting algorithm. However, as we move from single-dimensional space to planar space, there are two additional complications which are described in the following paragraphs. First of these complications is the *multitude of boundary conditions* and second one is the *multitude of unknown variables.*

[5]There is also a third improvement based approach: **Sequential Location Allocation method (SLA)**. This method has traditionally been used for discrete demand cases. In Appendix 4, we provide a continuous demand version.

In the single-dimensional case, this system of equations is considered as a *two-point boundary value problem* where initial boundary condition was the point of start of the market line and the terminal boundary condition was the ending point. For a given x-coordinate (y-coordinate) planar problem reduces to a line on the y-dimension (x-dimension), with two points as before. Therefore, when the $\mathcal{M}$ is a square shape, the boundary conditions are lines rather than points. In the previous section, we have made the assumption of a square-shaped market region $\mathcal{M}$, which translates into identical boundary points for every $x-$ and $y-$coordinate point. As pointed out earlier, with this shape assumption, there is no loss of generality of our results. For instance, when $\mathcal{M}$ is a trapezoid with parallel base and ceiling, then the boundary points at any $y-$coordinate level would be based on the affine relationship of non-parallel sides. Since we assume that we know the $\mathcal{M}$ in shape and size, then this affine relationship information would be available to determine the boundary values at any $y-$coordinate level.

The second complication with respect to the single dimensional case is that we need to determine an *allocation line* (BR) in the planar setting, rather than an *allocation point* as in the case of single-dimension. In other words, we have *an infinite number of unknowns* to solve for these system of differential equations, i.e. $A^x(y)$ for $\forall y \in Y_{BR}$ and $A^y(x)$ for $\forall x \in X_{BR}$. Overcoming this difficulty requires separate approaches for Manhattan and Euclidean-metric based distance measures. For Euclidean-metric cases, BR is a straight line. Since two points on a line is sufficient to define the affine relationship, we can reduce this problem down to a *two-point boundary value problem* with only two variables.

In what follows, we first derive the system of differential equations based

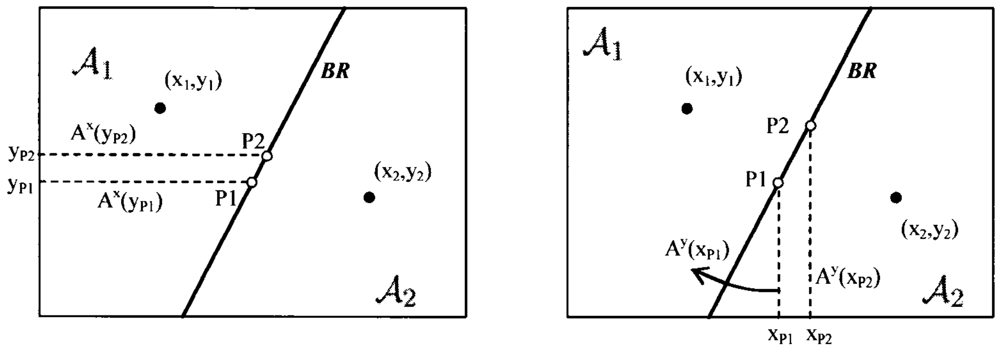

Figure 4.6: Choice of single dimensional allocation variables as the unknowns to the system of differential equations.

on the first order conditions of LAM-AVS, and then describe the constructive solution approach. Next, we provide a formal algorithm with an example application. Before the derivation of the system of differential equations, we need to decide on which of the two single dimensional-allocation variables, $A^x(y)$ or $A^y(x)$, we wish to denote as the unknowns. Figure 4.6 helps understanding the *equivalence* of choosing the unknown variables as either x-dimensional allocation variables ($A^x(y)$) or y-dimensional allocation variables ($A^y(x)$).

Figure 4.6 illustrates the two alternatives in characterizing the allocation line BR, i.e. deciding on whether to use $A^x(y)$ or $A^y(x)$ as the unknown variables. For both cases, we are choosing the same two points on the allocation line BR, namely **P1** and **P2**. Difference between $A^x(y)$ and $A^y(x)$ lies in the choice of the reference axis for measuring distance. First alternative (one on the left in Figure 4.6) measures location of $P1$ and $P2$ as distances from y-coordinate axis (i.e. x=0), thus the unknown variables are the x-dimensional allocation decisions ($A^x(y)$). Second alternative (one on the right in Figure 4.6) measures distance from x-coordinate axis (i.e. y=0), thus the unknown variables are y-dimensional allocation decisions ($A^y(x)$). In both cases, we need to specify two fixed-points on the reference axis chosen: (y_{P_1}, y_{P_2}) for $y - axis$ choice and (x_{P_1}, x_{P_2}) for $x - axis$ choice. Selection between these alternatives can be made arbitrarily since either of the two implies the same allocation line BR. Depending on the choice, the system of differential equations can be formed by applying the following first order conditions to the objective function of LAM-AVS.

$$\begin{aligned}\frac{dTC}{dA^x(y)} &= \frac{\partial TC}{\partial A^x(y)} + \frac{\partial TC}{\partial A^y(x)}\frac{\partial A^y(x)}{\partial A^x(y)} = 0 \qquad \text{for } A^x(y)\\ \frac{dTC}{dA^y(x)} &= \frac{\partial TC}{\partial A^y(x)} + \frac{\partial TC}{\partial A^x(y)}\frac{\partial A^x(y)}{\partial A^y(x)} = 0 \qquad \text{for } A^y(x)\end{aligned}$$

Hereafter, we choose $y - axis$ **as the reference axis** as such we will solve the first of the above differential equations for $A^x(y_{P1})$ and $A^x(y_{P2})$. We now provide the closed form expressions for these differential equations for two Euclidean-metric based distance measures, $p = L_2$ and L_2^2.

$$\frac{dTC}{dA^x(y)} = \frac{\partial TC}{\partial A^x(y)} + \frac{\partial TC}{\partial A^y(x)}\frac{\partial A^y(x)}{\partial A^x(y)} = 0 \tag{45}$$

$$d_p(\mathbf{x}_1^*, (A^x(y_{P1}), A^y(x))) - d_p(\mathbf{x}_2^*, (A^x(y_{P1}), A^y(x))) = 0 \tag{46}$$

$$d_p(\mathbf{x}_1^*, (A^x(y_{P2}), A^y(x))) - d_p(\mathbf{x}_2^*, (A^x(y_{P2}), A^y(x))) = 0 \tag{47}$$

These equations would take the following explicit forms for:

Euclidean-metric Case (L_2):

$$\begin{aligned}\sqrt{[(x_1^* - A^x(y_{P1}))^2 - (x_2^* - A^x(y_{P1}))^2]} - \sqrt{[(y_2^* - y_{P1})^2 - (y_1^* - y_{P1})^2]} &= 0\\ \sqrt{[(x_1^* - A^x(y_{P2}))^2 - (x_2^* - A^x(y_{P2}))^2]} - \sqrt{[(y_2^* - y_{P2})^2 - (y_1^* - y_{P2})^2]} &= 0\end{aligned}$$

Squared Euclidean-metric Case (L_2^2):

$$
\begin{aligned}
\left[(x_1^* - A^x(y_{P1}))^2 - (x_2^* - A^x(y_{P1}))^2\right] - \left[(y_2^* - y_{P1})^2 - (y_1^* - y_{P1})^2\right] &= 0 \\
\left[(x_1^* - A^x(y_{P2}))^2 - (x_2^* - A^x(y_{P2}))^2\right] - \left[(y_2^* - y_{P2})^2 - (y_1^* - y_{P2})^2\right] &= 0
\end{aligned}
$$

where $\mathbf{x}_1^* = (x_1^*, y_1^*)$ and $\mathbf{x}_2^* = (x_2^*, y_2^*)$ are the coordinates of the optimal facility locations of allocation decisions, $\mathcal{A}_1$ and $\mathcal{A}_2$, implied by $A^x(y_{P1})$ and $A^x(y_{P2})$.

In solving the above system of equations as a boundary-value problem, $A^x(y_{P1})$ and $A^x(y_{P2})$ are considered as the system's state variables. The origin ($x = 0$) and the market region boundary ($x{=}M$) are the two boundary values defining the initial and ending values of the system's state. Shooting algorithm, as described before in the single-dimensional case, starts with two arbitrary and independent triggers ($A^x(y_{P1})$ and $A^x(y_{P2})$) at the system's initial state (boundary value). Then, the procedure sequentially decides on the allocation decisions satisfying the state transition equations (45) until the end, i.e., after deciding on the final single dimensional allocation decision. If these initial triggers are not correct, then the ending state value would differ from the boundary value (M). As the accuracy of the initial triggers are improved, the system gets closer to the boundary value at the end. From this point onward, we denote this *variable* ending state value as M' and refer M as the actual ending boundary value of square market region $\mathcal{M}$. We now describe the constructive solution approach, i.e. the shooting algorithm, in more detail by pointing out the differences from the single dimensional case.

For ease of exposition, let's define the following notation as illustrated in

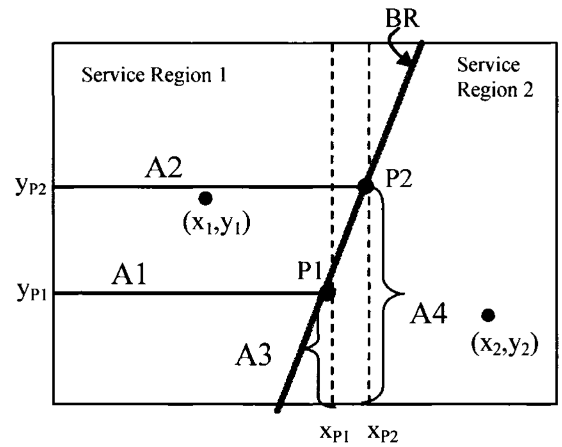

Figure 4.7: Initial triggers for the constructive solution approach (Euclidean-metric based measures)

Figure 4.7.

$$A1 \ : \ = A^x(y_{P1}) \text{ and } A2 := A^x(y_{P2})$$
$$A3 \ : \ = A^y(x_{P1}) \text{ and } A4 := A^y(x_{P2})$$

As in the single-dimensional case application of the shooting algorithm, we first determine two initial triggers, $A1$ and $A2$, at pre-specified points, y_{P1} and y_{P2}. These two triggers determine the straight allocation line (BR) passing through $(A1, y_{P1})$ and $(A2, y_{P2})$, thus the two service regions, $\mathcal{A}_1$ and $\mathcal{A}_2$. While some of the triggers, $A1$ and $A2$, fully characterize $\mathcal{A}_1$, some do not. In both cases, $\mathcal{A}_2$ is allowed to be a variable region such that its right-hand side (M') is not fixed. In order to further illustrate this, two alternative cases are plotted in the Figure 4.8. In the first case (on the left), $\mathcal{A}_1$, **service region 1**, is defined fully when BR is constructed, but $\mathcal{A}_2$'s right-hand side border line M' (i.e. ending state), which is conjectured to be a vertical line, is not. In the

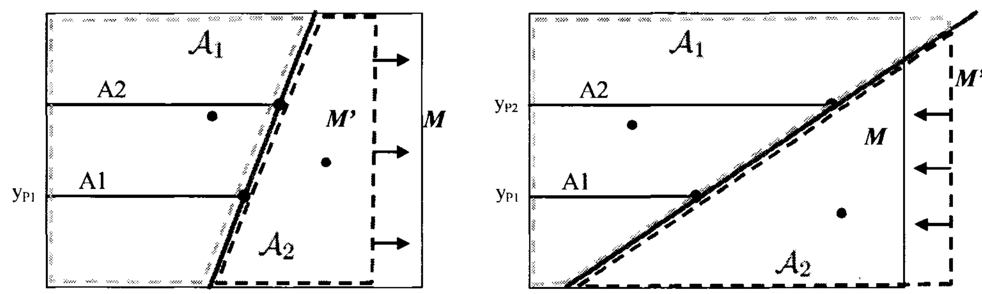

Figure 4.8: Illustration of cases where initial single-dimensional allocation triggers characterize only one or both of the service regions.

second case, since the M' is part of both $\mathcal{A}_1$ and $\mathcal{A}_2$, neither of them are fully characterized with the determination of BR. In other words, the two initial triggers $A1$ and $A2$ are sufficient to determine BR, but not to determine the service regions $\mathcal{A}_1$ and $\mathcal{A}_2$ at all times. In either case, the ultimate goal of choosing initial triggers is to coincide M' with M.

In the single dimensional case, the initial trigger allowed us to determine the allocation boundaries on the market region (i.e. market line) and identify a single boundary value M'. In the planar case, we first decide on the initial triggers ($A1$ and $A2$ in Figure 4.8) to determine the allocation line BR, then we parametrize the $\mathcal{A}_1$ and $\mathcal{A}_2$ based on the BR and the ending state M'. Note that this parametrization is analogous to the tiling operation in the single dimensional case. This parametrization works as follows: Optimal locations of facilities, (x_1^*, y_1^*) and (x_2^*, y_2^*), are expressed in terms of M' and used to solve the differential equations in (45) for M'. Since there is only one unknown but two equations, (46) and (47), there would be two different solutions of M' (i.e. $M1'$ and $M2'$), one for each initial trigger. When these initial triggers ($A1$ and $A2$) are optimal, $M1'$ and $M2'$ would be equal to the market boundary, i.e. $M1' = M$ and $M2' = M$. This is the difference of the two dimensional constructive approach from the single dimensional equivalent, which had one ending state for a single initial trigger. Also note that (x_1^*, y_1^*) and (x_2^*, y_2^*) are functions of $(A1, A2)$ and M' which satisfy the below locational optimality condition as well as the system of equations in (48) and (49) for Euclidean-metric and Squared Euclidean-metric cases, respectively.

$$\mathbf{x}_i^* = \arg\min_{(\mathbf{x}_i)} \int_{\mathcal{A}_i(M')} d_p(\mathbf{x}_i, \mathbf{x}) D(\mathbf{x}) d\mathbf{x} \qquad \text{for } i = 1, 2 \text{ and } p = L_2, L_2^2$$

where $\mathcal{A}_i(M')$ is used to illustrate the dependence of the service region on the ending boundary condition as illustrated in Figure 4.8.

Euclidean-metric Case (L_2):

$$\begin{aligned} \sqrt{[(x_1^* - A1))^2 - (x_2^* - A1))^2]} - \sqrt{[(y_2^* - A3)^2 - (y_1^* - A3)^2]} &= 0 \quad (48) \\ \sqrt{[(x_1^* - A2)^2 - (x_2^* - A2))^2]} - \sqrt{[(y_2^* - A4)^2 - (y_1^* - A4)^2]} &= 0 \end{aligned}$$

Squared Euclidean-metric Case (L_2^2):

$$\begin{aligned} \left[(x_1^* - A1))^2 - (x_2^* - A1))^2\right] - \left[(y_2^* - A3)^2 - (y_1^* - A3)^2\right] &= 0 \quad (49) \\ \left[(x_1^* - A2)^2 - (x_2^* - A2))^2\right] - \left[(y_2^* - A4)^2 - (y_1^* - A4)^2\right] &= 0 \end{aligned}$$

Thus the optimal locations depend not only on the *initial triggers,* but also on the *ending boundary condition.*

$$(x_1^*, y_1^*) \quad \propto \quad (A1, A2, M') \tag{50}$$

$$(x_2^*, y_2^*) \quad \propto \quad (A1, A2, M') \tag{51}$$

The shooting algorithm tries to minimize the deviation from the pre-specified boundary condition M; in other words it aims to find the root of absolute difference between M and M'. Thus, in a similar approach to the single dimensional case, we here define the root finding functional as below.

$$\digamma(A1, A2) = (\digamma_1(A1, A2), \digamma_2(A1, A2))^T = \begin{pmatrix} \digamma_1(A1, A2) = M1' - M \\ \digamma_2(A1, A2) = M2' - M \end{pmatrix}$$

where

$M1'$:is the ending state M' which solves the first equation in (48) or (49)

$M2'$:is the ending state M' which solves the second equation in (48) or (49)

The objective is to find the root of $\digamma(A1, A2) = (0, 0)$. In solving this root-finding problem, we could use either the first-order line-search methods or second-order methods. In the case of *Euclidean-metric(L_2)*, optimal locations given the allocation decisions is computationally demanding, thus we will use a first-order method with a fixed step size. For the *Squared Euclidean-metric(L_2^2)* measure, we will use the *Newton-Raphson multidimensional root-finding method* since optimal locations are significantly easier to solve for. Newton-Raphson is also preferable because of its quadratic local convergence property while assuming that our initial triggers are sufficiently good. The

Newton-Raphson method uses the following iteration formula to calculate the next candidate for the root.

$$\begin{pmatrix} A1 \\ A2 \end{pmatrix}^{new} = \begin{pmatrix} A1 \\ A2 \end{pmatrix}^{old} - J^{-1}\mathcal{F} \tag{52}$$

where

J, J^{-1} : are the Jacobian and inverse Jacobian matrix of $\mathcal{F}$, i.e. $J_{ij} = \frac{\partial \mathcal{F}_i}{\partial A_j}$

Analytical calculation of J is impractical since the closed form expression of $\mathcal{F}(A1, A2)$ is not available. Hence, as in the single dimensional case, we utilize a discrete approximation which is the centered difference approximation of the first-order differential $\frac{\partial \mathcal{F}_i}{\partial A_j}$.

$$\frac{\partial \mathcal{F}_1}{\partial A_1} \simeq \frac{\mathcal{F}_1(A1+h, A2) - \mathcal{F}_1(A1-h, A2)}{2h}$$

$$\frac{\partial \mathcal{F}_1}{\partial A_2} \simeq \frac{\mathcal{F}_1(A1, A2+h) - \mathcal{F}_1(A1, A2-h)}{2h}$$

$$\frac{\partial \mathcal{F}_2}{\partial A_1} \simeq \frac{\mathcal{F}_2(A1+h, A2) - \mathcal{F}_2(A1-h, A2)}{2h}$$

$$\frac{\partial \mathcal{F}_2}{\partial A_2} \simeq \frac{\mathcal{F}_2(A1, A2+h) - \mathcal{F}_2(A1, A2-h)}{2h}$$

As mentioned previously, different initial triggers, A_1 and A_2, create different scenarios for the two service regions $\mathcal{A}_1$ and $\mathcal{A}_2$. As illustrated in Figure 4.8, some of these initial trigger combinations will fully characterize $\mathcal{A}_1$ and some will not. These scenarios are dependent on the type of BR characterized by the initial triggers and displayed in Figure 4.9.

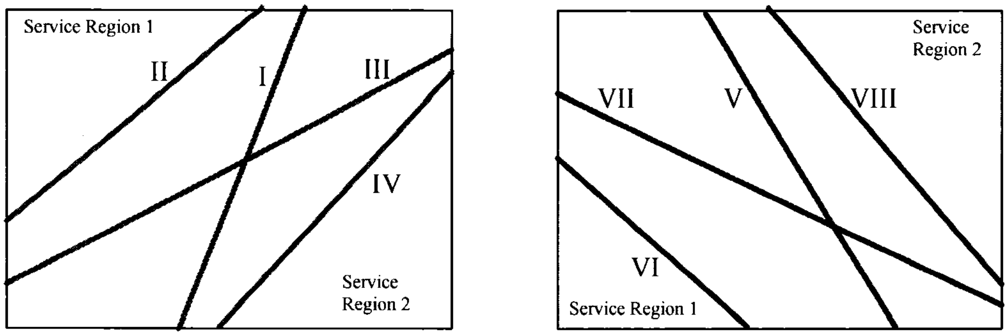

Figure 4.9: Allocation line scenarios.

Now we provide a formal shooting algorithm for our basic model:

Shooting Algorithm: Euclidean-metric Based Distance Measure (L_2 and L_2^2)

Step 1. Define and Initialize the model parameters and variables

j : index for the feasibility iterations (i.e. $j^* = \{j | \ \epsilon_{BOUND} \geq |M1' - M|$ and $\epsilon_{BOUND} \geq |M2' - M|\}$

ϵ_{BOUND} : epsilon parameter for feasibility stopping decision

h : centered difference approximation parameter for partial differentials

$A1^j$: j^{th} iteration estimate for the first service region size at y_{P1}

$A2^j$: j^{th} iteration estimate for the first service region size at y_{P2}

$\mathbf{x}_i^* = (x_i^*, y_i^*)$: optimal locations corresponding to $\mathcal{A}_{i=1,2}^j$

$(M')^j$: boundary variable (i.e. $M^j = M$ is the feasible boundary condition)

$M1', M2'$: solutions for M^j to the first and second equations in (48) or (49), respectively.

Set $j = 0$, $A1^{j=1}$ and $A2^{j=1}$

Step 2. Update the first service region sizes $A1^j$ and $A2^j$

Do While ($|M1' - M| \geq \epsilon_{BOUND}$ and $|M2' - M| \geq \epsilon_{BOUND}$):

$j = j + 1$

Step 2.1. Calculate Slope(a) and Intercept(b) of BR using $\left(\frac{y-A3}{A3-A4}\right) = \left(\frac{x-A1}{A1-A2}\right)$

Slope $\quad a := \frac{A3-A4}{A1-A2}$

Intercept $\quad b := \frac{A1A4-A3A2}{A1-A2}$

Step 2.2. Parameterize $\mathcal{A}_1$ and $\mathcal{A}_2$ as a function of $(M')^j$

Type I: If $a \geq 0$ and $br(x = 100) \geq M$ and $br(x = 0) \leq 0$ then

$\mathcal{A}_1 := \{(x, y) | y \in Y_{BR} \text{ and } x \in [0, br(y)^{-1}]\}$

$\mathcal{A}_2 := \{(x, y) | y \in Y_{BR} \text{ and } x \in [br(y)^{-1}, (M')^j]\}$

Type II: If $a \geq 0$ and $br(x = 100) \geq M$ and $br(x = 0) \geq 0$ then

$\mathcal{A}_1 := \{(x, y) | y \in Y_{BR} \text{ and } x \in [0, br(y)^{-1}]\}$

$\mathcal{A}_2 := \{(x, y) | y \in Y_{BR} \text{ and } x \in [br(y)^{-1}, (M')^j] \cup y \in Y_{A2} \text{ and } x \in [0, (M')^j]\}$

Type III: If $a \geq 0$ and $br(x = 100) \leq M$ and $br(x = 0) \geq 0$ then

$\mathcal{A}_1 := \{(x, y) | y \in Y_{BR} \text{ and } x \in [0, br(y)^{-1}] \cup y \in Y_{A1} \text{ and } x \in [0, (M')^j]\}$

$\mathcal{A}_2 := \{(x, y) | y \in Y_{BR} \text{ and } x \in [br(y)^{-1}, (M')^j] \cup y \in Y_{A2} \text{ and } x \in [0, (M')^j]\}$

Type IV: If $a \geq 0$ and $br(x = 100) \leq M$ and $br(x = 0) \leq 0$ then

$\mathcal{A}_1 := \{(x, y) | y \in Y_{BR} \text{ and } x \in [0, br(y)^{-1}] \cup y \in Y_{A1} \text{ and } x \in [0, (M')^j]\}$

$\mathcal{A}_2 := \{(x, y) | y \in Y_{BR} \text{ and } x \in [br(y)^{-1}, (M')^j]\}$

Type V: If $a \leq 0$ and $br(x = 100) \leq 0$ and $br(x = 0) \geq M$ then

$\mathcal{A}_1 := \{(x, y) | y \in Y_{BR} \text{ and } x \in [0, br(y)^{-1}]\}$

$\mathcal{A}_2 := \{(x, y) | y \in Y_{BR} \text{ and } x \in [br(y)^{-1}, (M')^j]\}$

Type VI: If $a \leq 0$ and $br(x = 100) \leq 0$ and $br(x = 0) \leq M$ then

$\mathcal{A}_1 := \{(x,y)|y \in Y_{BR} \text{ and } x \in [0, br(y)^{-1}]\}$

$\mathcal{A}_2 := \{(x,y)|y \in Y_{BR} \text{ and } x \in [br(y)^{-1}, (M')^j] \ \cup \ y \in Y_{A2} \text{ and } x \in [0, (M')^j]\}$

Type VII: If $a \leq 0$ and $br(x = 100) \geq 0$ and $br(x = 0) \leq M$ then

$\mathcal{A}_1 := \{(x,y)|y \in Y_{BR} \text{ and } x \in [0, br(y)^{-1}] \cup \ y \in Y_{A1} \text{ and } x \in [0, (M')^j]\}$

$\mathcal{A}_2 := \{(x,y)|y \in Y_{BR} \text{ and } x \in [br(y)^{-1}, (M')^j] \ \cup \ y \in Y_{A2} \text{ and } x \in [0, (M')^j]\}$

Type VIII: If $a \leq 0$ and $br(x = 100) \geq 0$ and $br(x = 0) \geq M$ then

$\mathcal{A}_1 := \{(x,y)|y \in Y_{BR} \text{ and } x \in [0, br(y)^{-1}] \cup \ y \in Y_{A1} \text{ and } x \in [0, (M')^j]\}$

$\mathcal{A}_2 := \{(x,y)|y \in Y_{BR} \text{ and } x \in [br(y)^{-1}, (M')^j]\}$

Step 2.3. Solve the following single facility location problems as a function of $(M')^j$

$$\mathbf{x}_1^* = (x_1^*, y_1^*) := \arg \min_{\mathbf{x}_1 = (x_1, y_1)} (\textstyle\int_{\mathcal{A}_1} d_p(\mathbf{x}_1, \mathbf{x}) D(\mathbf{x}) d\mathbf{x})$$
$$\mathbf{x}_2^* = (x_2^*, y_2^*) := \arg \min_{\mathbf{x}_2 = (x_2, y_2)} (\textstyle\int_{\mathcal{A}_2} d_p(\mathbf{x}_2, \mathbf{x}) D(\mathbf{x}) d\mathbf{x})$$

Step 2.4. Solve following for the boundary value $M1^j$

$$d_p(\mathbf{x}_1^*, (A1^j, A3)) - d_p(\mathbf{x}_2^*, (A1^j, A3)) = 0$$

Step 2.5. Solve following for the boundary value $M2^j$

$$d_p(\mathbf{x}_1^*, (A2^j, A4)) - d_p(\mathbf{x}_2^*, (A2^j, A4)) = 0$$

Step 2.6. Calculate $\mathcal{F}_1$ and $\mathcal{F}_2$

$$\mathcal{F}_1(A1^j, A2^j) = M1' - M$$

$$\mathcal{F}_2(A1^j, A2^j) = M2' - M$$

Step 2.7. Approximate J using centered difference approximation (only for the *Squared Euclidean – metric case*, i.e. L_2^2)

Repeat Steps 2.1-2.6 for $(A1^j - h, A2^j)$, $(A1^j + h, A2^j)$, $(A1^j, A2^j -$

$h),(A1^j, A2^j + h)$

Step 2.8. Assign

$Euclidean - metric\ (L_2)$

$$\begin{pmatrix} A1 \\ A2 \end{pmatrix}^{j+1} = \begin{pmatrix} A1 \\ A2 \end{pmatrix}^{j} - \lambda \begin{pmatrix} \digamma_1(A1^j, A2^j) \\ \digamma_2(A1^j, A2^j) \end{pmatrix}$$

$Squared\ Euclidean - metric\ (L_2^2)$

$$\begin{pmatrix} A1 \\ A2 \end{pmatrix}^{j+1} = \begin{pmatrix} A1 \\ A2 \end{pmatrix}^{j} - J^{-1} \begin{pmatrix} \digamma_1(A1^j, A2^j) \\ \digamma_2(A1^j, A2^j) \end{pmatrix}$$

where $J = \begin{bmatrix} \frac{\partial \digamma_1}{\partial A1} & \frac{\partial \digamma_1}{\partial A2} \\ \frac{\partial \digamma_2}{\partial A1} & \frac{\partial \digamma_2}{\partial A2} \end{bmatrix}$ and λ is the constant step length

Return.

Step 3. Terminate with the solution $A1^j$, $A2^j$ and BR

Before we illustrate the application of the above constructive solution approach, it is important to distinguish between the Squared Euclidean-metric and Euclidean-metric. Recalling the first-order conditions of the previous section, we have the following for two metrics.

$$\mathbf{x}_i^* = \arg\min_{(\mathbf{x}_i)} \int_{\mathcal{A}_i(M')} d_p(\mathbf{x}_i, \mathbf{x}) D(\mathbf{x}) d\mathbf{x} \qquad for\ i = 1, 2$$

$$\mathbf{x}_i^* = \frac{\int_{\mathcal{A}_i} \mathbf{x} D(\mathbf{x}) d\mathbf{x}}{\int_{\mathcal{A}_i} D(\mathbf{x}) d\mathbf{x}} \quad \text{for Squared Euclidean-metric, i.e. } p = L_2^2 \tag{53}$$

$$\int_{\mathcal{A}_i} \frac{(\mathbf{x}_i^* - \mathbf{x})}{||\mathbf{x}_i^* - \mathbf{x}||} = 0 \quad \text{for Euclidean-metric, i.e. } p = L_2 \tag{54}$$

Whereas analytical solution to (53) is straightforward, same is not true for (54). In order to find the solution satisfying (54), we need to use the Weiszfeld's approach (Weiszfeld, 1937). This requires minor modification to the constructive algorithm presented above, namely repetitive iterations of the algorithmic Steps 2.3 to 2.6 with variable M'. Since we cannot solve for M' and optimal locations at the same time, we start with an initial estimate of M' and find the optimal locations based on the standard Weiszfeld procedure. Note that we do not need to account for the singularity due to continuous demand distribution. We then update the M' estimate to satisfy the the equidistance property, i.e. optimality condition of LAM-AVS presented in the previous section. Next, we provide examples for both distance measures to illustrate the differences.

Note that in the implementation of the shooting algorithm we are assuming that our shooting levels $y_{P1} = A3$ and $y_{P2} = A4$ are in the set of Y_{BR} at the optimal solution.[6]

Example 4.1.: Constructive Solution Approach- Squared Euclidean-metric (L_2^2) Case

[6] If we start with such levels outside the Y_{BR} or X_{BR} of the optimal solution, counstructive solution would not converge. This would then indicate us to either change the shooting levels or switch to horizontal shooting or to vertical shooting pattern. An example of this switching is presented in the example for L_2.

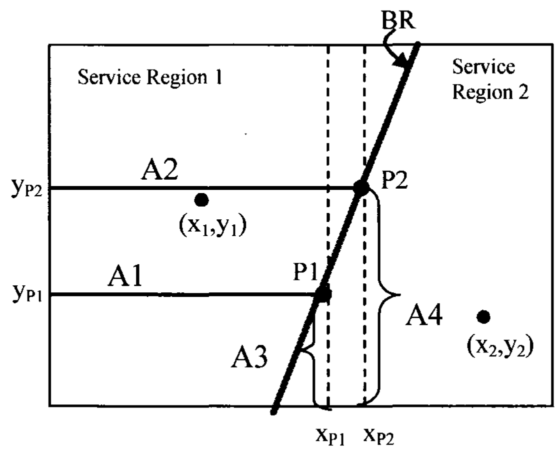

Figure 4.10: Illustration of shooting algorithm using initial triggers, A1 and A2.

For convenience, the notation for the parameters and variables are illustrated in Figure 4.10. Here we have a square-shaped market region $\mathcal{M}$=$\{(x, y)|x \in (0, 100)$ and $y \in (0, 100)\}$, i.e. $M = 100$. We wish to determine an optimal allocation decision for a linear demand density function $(D(x, y) = 100+10x+5y)$ over the market region $\mathcal{M}$. The starting solution for this instance is $A1$=30 and $A2$ =45 at $y_{P1} = A3 = 40$ and $y_{P2} = A4 = 50$, respectively.

The steps of the algorithm is as follows:

ITERATION 1

Step 1. Initialize the model parameters and variables

Set: $j = 1$

$y_{P1} = A3 = 40$ and $y_{P2} = A4 = 50$

$A1^{j=1}$=30 and $A2^{j=1}$ =45

$\epsilon_{BOUND} = 0.01$ and $h = 0.1$

$M = 100$

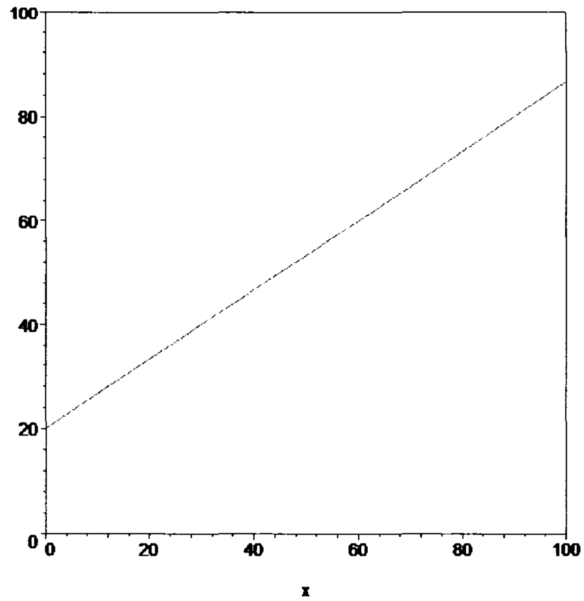

Figure 4.11: Allocation decision at the start of iteration 1 (Example 4.1).

Step 2. Update the first service region sizes $A1^{j=1}$ and $A2^{j=1}$.

Step 2.1. Calculate Slope(a) and Intercept(b) of BR using $\left(\frac{y-A3}{A3-A4}\right) = \left(\frac{x-A1^{j=1}}{A1^{j=1}-A2^{j=1}}\right)$

Slope $a := \frac{2}{3}$, Intercept $b := 20$

$br(x) = y = \frac{2}{3}x + 20$, $br(y)^{-1} = x = \frac{y-20}{2/3}$

$Y_{BR} = [20, \frac{260}{3}]$, $X_{BR} = [0, 100]$, $Y_{A1} = [\frac{260}{3}, 100]$, $Y_{A2} = [0, 20]$

Step 2.2. Parametrize $\mathcal{A}_1$ and $\mathcal{A}_2$ in terms of $(M')^{j=1}$

Type III: If $a \geq 0$ and $br(x = 100) = 260/3 \leq M = 100$ and $br(x = 0) = 20 \geq 0$ then

$\mathcal{A}_1 := \{(x, y)|y \in Y_{BR} = [20, \frac{260}{3}]$ and $x \in [0, \frac{y-20}{2/3}] \cup y \in Y_{A1} = [\frac{260}{3}, 100]$ and $x \in [0, (M')^{j=1}]\}$

$\mathcal{A}_2 := \{(x, y)|y \in Y_{BR} = [20, \frac{260}{3}]$ and $x \in [\frac{y-20}{2/3}, (M')^{j=1}] \cup y \in Y_{A2} = [0, 20]$ and $x \in [0, (M')^{j=1}]\}$

Step 2.3. Solve the following single facility location problems in terms of

$(M')^{j=1}$

$$x_1^* = .66667\left(M'\right)^{j=1} - 16.000 - \frac{769.30}{(M')^{j=1}-192.64} + \frac{769.30}{(M')^{j=1}+64.072}$$
$$y_1^* = 68.002 + \frac{64.236}{(M')^{j=1}-192.64} - \frac{64.236}{(M')^{j=1}+64.072}$$
$$x_2^* = .66665\left(M'\right)^{j=1} - 2.9996 + \frac{8998.7}{2.5926\left(\left(M'\right)^{j=1}\right)^2+166.67(M')^{j=1}+2999.9}$$
$$y_2^* = 14.000\left(M'\right)^{j=1} - \frac{8665.6}{2.5926\left(\left(M'\right)^{j=1}\right)^2+166.67(M')^{j=1}+2999.9}$$

Step 2.4. Solve following for the boundary value $(M1')^{j=1}$

$$[(x_1^* - 30)^2 - (x_2^* - 30)^2] - [(y_2^* - 40)^2 - (y_1^* - 40)^2] = 0$$

$(M1')^{j=1}$ =101.682

Step 2.5. Solve following for the boundary value $(M2')^{j=1}$

$$[(x_1^* - 45)^2 - (x_2^* - 45)^2] - [(y_2^* - 50)^2 - (y_1^* - 50)^2] = 0$$

$(M2')^{j=1}$ =95.718

Step 2.6. Calculate F_1 and F_2

F_1 =101.682-100=1.682

F_2 =95.718-100=-4.282

Step 2.7. Approximate J using centered difference approximation

Repeat Steps 2.1-2.6 for $(A1^{j=1} - 0.1, A2^{j=1})$, $(A1^{j=1} + 0.1, A2^{j=1})$, $(A1^{j=1}, A2^{j=1} - 0.1)$, and $(A1^{j=1}, A2^{j=1} + 0.1)$

For $(A1^{j=1} - h, A2^{j=1})$: F_1 =1.7569302 and F_2 =-4.24153533

For $(A1^{j=1} + h, A2^{j=1})$: F_1 =1.6067966 and F_2 =-4.32283664

For $(A1^{j=1}, A2^{j=1} - h)$: F_1 =1.5437098 and F_2 =-4.39969988

For $(A1^{j=1}, A2^{j=1} + h)$: F_1 =1.8196613 and F_2 =-4.16472687

Step 2.8. Assign

$$\begin{pmatrix} A1 \\ A2 \end{pmatrix}^{j=2} = \begin{pmatrix} 30 \\ 45 \end{pmatrix}^{j=1} - J^{-1}\begin{pmatrix} 1.682 \\ -4.282 \end{pmatrix} = \begin{pmatrix} 54.558 \\ 57.142 \end{pmatrix}$$

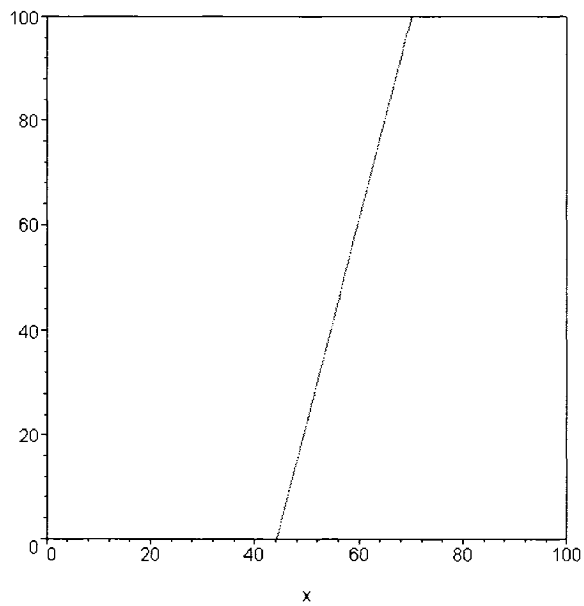

Figure 4.12: Allocation decision at the start of iteration 2 (Example 4.1).

$$\text{where } J = \begin{bmatrix} \frac{\partial F_1}{\partial A1} = -0.750668 & \frac{\partial F_1}{\partial A2} = 1.379757 \\ \frac{\partial F_2}{\partial A1} = -0.406507 & \frac{\partial F_2}{\partial A2} = 1.174865 \end{bmatrix}$$

ITERATION 2

Set $j = 2$, $A1^{j=2}$=54.558 and $A2^{j=2} = 57.142$

Step 2.1. Calculate Slope(a) and Intercept(b) of BR using $\left(\frac{y-A3}{A3-A4}\right) = \left(\frac{x-A1^{j=2}}{A1^{j=2}-A2^{j=2}}\right)$

Slope $a := 3.871$, Intercept $b := -171.197$

$br(x) = y = 3.871x - 171.197$, $br(y)^{-1} = x = \frac{y+171.197}{3.871}$

$Y_{BR} = [0, 100]$, $X_{BR} = [44.23, 70.06]$, $X_{A1} = [0, 44.23]$, $X_{A2} = [44.23, 100]$

Step 2.2. Parametrize $\mathcal{A}_1$ and $\mathcal{A}_2$ in terms of $(M')^{j=2}$

Type I: If $a \geq 0$ and $br(x = 100) = 215.903 \geq M$ and $br(x = 0) = -171.196 \leq 0$ then

$$\mathcal{A}_1 := \{(x, y) | y \in Y_{BR} = [0, 100] \text{ and } x \in [0, \frac{y+171.197}{3.871}]\}$$

$$\mathcal{A}_2 := \{(x,y) | y \in Y_{BR} = [0,100] \text{ and } x \in [\tfrac{y+171.197}{3.871}, (M')^{j=2}]\}$$

Step 2.3. Solve the following single facility location problems in terms of $(M')^{j=2}$

$x_1^* = 34.4044$

$y_1^* = 61.5831$

$x_2^* = 35.000000 - \frac{742.95692}{(M')^{j=2} - 58.60036164} - \frac{1707.0429}{(M')^{j=2} + 128.6003616}$

$y_2^* = 50.000000 - \frac{205.43032}{(M')^{j=2} - 58.60036164} + \frac{1038.7637}{(M')^{j=2} + 128.6003616}$

Step 2.4. Solve following for the boundary value $(M1')^{j=2}$

$[(x_1^* - 54.558)^2 - (x_2^* - 54.558)^2] - [(y_2^* - 40)^2 - (y_1^* - 40)^2] = 0$

$(M1')^{j=2} = 104.091$

Step 2.5. Solve following for the boundary value $(M2')^{j=2}$

$[(x_1^* - 57.142)^2 - (x_2^* - 57.142)^2] - [(y_2^* - 50)^2 - (y_1^* - 50)^2] = 0$

$(M2')^{j=2} = 104.612$

Step 2.6. Calculate $\mathcal{F}_1$ and $\mathcal{F}_2$

$\mathcal{F}_1 = 104.091 - 100.0 = 4.091$

$\mathcal{F}_2 = 104.612 - 100.0 = 4.612$

Step 2.7. Approximate J using centered difference approximation

Repeat Steps 2.1-2.6 for $(A1^{j=2} - h, A2^{j=2})$, $(A1^{j=2} + h, A2^{j=2})$, $(A1^{j=2}, A2^{j=2} - h)$, and $(A1^{j=2}, A2^{j=2} + h)$

For $(A1^{j=2} - h, A2^{j=2})$: $\mathcal{F}_1 = 4.0317955$ and $\mathcal{F}_2 = 4.6281418$

For $(A1^{j=2} + h, A2^{j=2})$: $\mathcal{F}_1 = 4.1514577$ and $\mathcal{F}_2 = 4.5943379$

For $(A1^{j=2}, A2^{j=2} - h)$: $\mathcal{F}_1 = 4.0112743$ and $\mathcal{F}_2 = 4.4369477$

For $(A1^{j=2}, A2^{j=2} + h)$: $\mathcal{F}_1 = 4.1702835$ and $\mathcal{F}_2 = 4.7847247$

Step 2.8. Assign

$$\begin{pmatrix} A1 \\ A2 \end{pmatrix}^{j=3} = \begin{pmatrix} 54.558 \\ 57.142 \end{pmatrix}^{j=2} - J^{-1} \begin{pmatrix} 4.091 \\ 4.612 \end{pmatrix} = \begin{pmatrix} 51.624 \\ 54.205 \end{pmatrix}$$

$$\text{where } J = \begin{bmatrix} \frac{\partial F_1}{\partial A1} = .5983110 & \frac{\partial F_1}{\partial A2} = .7950460 \\ \frac{\partial F_2}{\partial A1} = -.1690195 & \frac{\partial F_2}{\partial A2} = 1.738885 \end{bmatrix}$$

When we continue with the **iterations 3** and **4**, we would obtain the following local optimum solution.

$$\begin{pmatrix} A1 \\ A2 \end{pmatrix}^{j=5} = \begin{pmatrix} 51.569216 \\ 54.181516 \end{pmatrix}$$

$$\begin{aligned} \mathbf{x}_1^* &= (x_1^*, y_1^*) = (32.662, 61.992) \\ \mathbf{x}_2^* &= (x_2^*, y_2^*) = (78.816, 49.935) \end{aligned}$$

$$\begin{aligned} br(x) &: \; y = 3.828045x - 157.409263 \\ TC &= 8,459,944,237.00 \end{aligned}$$

Complete results for this starting point displayed in Table 4.1. Resulting allocation decision is illustrated in Figure 4.13. Observe that converge is attained when $|M' - M| \leq \in_{BOUND} = 0.01$.

Iteration(j)	A1	A2	A3	A4	Slope	Intercept	Type	M1'	M2'	F_1	F_2
1	30.000	45.000	40	50	0.667	20.000	III	101.68	95.72	1.6820	-4.2820
2	54.558	57.142	40	50	3.871	-171.197	I	104.09	104.61	4.0910	4.6120
3	51.624	54.205	40	50	3.875	-160.062	I	100.04	100.03	0.0418	0.0261
4	51.569	54.182	40	50	3.828	-157.427	I	100.00	100.00	0.0001	0.0000

Table 4.1. Constructive solution algorithm iteration results for initial triggers $A_1 = 30$ and $A_2 = 45$ with L_2^2 (Example 4.1).

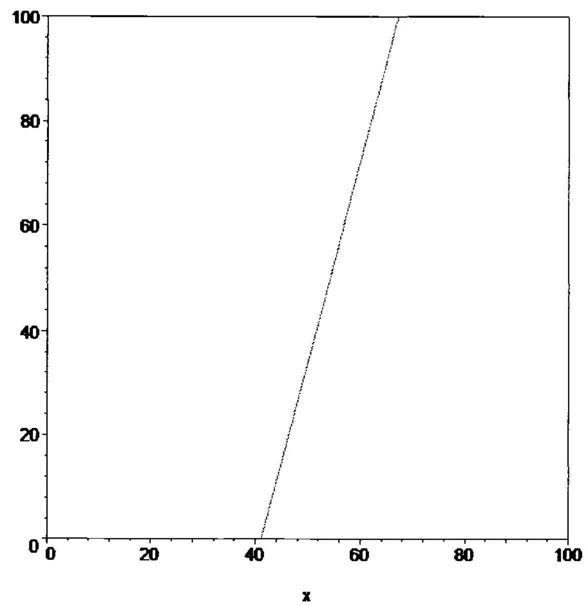

Figure 4.13: Allocation decision in the end of iteration 4 (Example 4.1).

To illustrate the existence of **multiple local solutions**, Table 4.2. displays the iteration results when constructive algorithm is started with the initial triggers $A1$=35 and $A2$ =40 at $y_{P1} = A3 = 40$ and $y_{P2} = A4 = 50$, respectively.

Iteration(j)	A1	A2	A3	A4	Slope	Intercept	Type	M1'	M2'	F_1	F_2
1	35.000	40.000	40	50	2.000	-30.000	I	84.98	82.10	-15.0236	-17.9038
2	44.574	52.467	40	50	1.267	-16.470	I	99.35	99.05	-0.6536	-0.9537
3	45.487	53.280	40	50	1.283	-18.369	I	100.01	100.01	0.0108	0.0129
4	45.479	53.270	40	50	1.284	-18.374	I	100.00	100.00	0.0000	0.0000

Table 4.2. Constructive solution algorithm iteration results for initial triggers $A_1 = 35$ and $A_2 = 40$ with L_2^2 (Example 4.2).

The solution to this problem and the objective function value is as below and Figure 4.14 illustrates the corresponding allocation decisions

$$\begin{pmatrix} A1 \\ A2 \end{pmatrix}^{j=5} = \begin{pmatrix} 45.4787 \\ 53.2697 \end{pmatrix}$$

$$\begin{aligned} \mathbf{x}_1^* &= (x_1^*, y_1^*) = (39.664, 70.593) \\ \mathbf{x}_2^* &= (x_2^*, y_2^*) = (76.565, 41.844) \end{aligned}$$

$$\begin{aligned} br(x) &: y = 1.2835x - 18.3738 \\ TC &= 8,534,652,763.00 \end{aligned}$$

We have conducted a large number of experiments and the best yet solution with this example, starting with initial triggers $A1$=10 and $A2$ =50 at $y_{P1} = A3 = 45$ and $y_{P2} = A4 = 50$, is as follows.

$$\begin{pmatrix} A1 \\ A2 \end{pmatrix}^{j=6} = \begin{pmatrix} 10.27147 \\ 43.31926 \end{pmatrix}$$

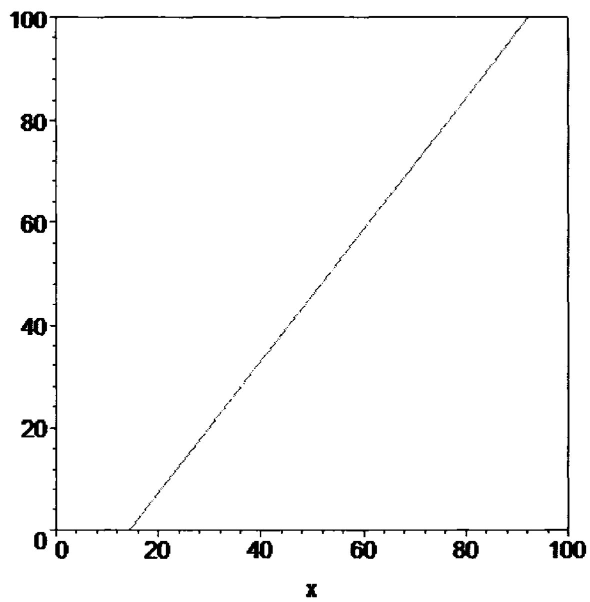

Figure 4.14: Allocation decision with the starting solution $A_1 = 35$ and $A_2 = 40$ (Example 4.2).

$$\mathbf{x}_1^* = (x_1^*, y_1^*) = (56.4500, 77.0701)$$
$$\mathbf{x}_2^* = (x_2^*, y_2^*) = (63.8702, 28.0258)$$

$$br(x) \ : \ y = 0.1512961x + 43.44597$$
$$TC = 7,965,251,223.00$$

Table 4.3. displays the iteration results. Figure 4.15 illustrates the corresponding allocation decision.

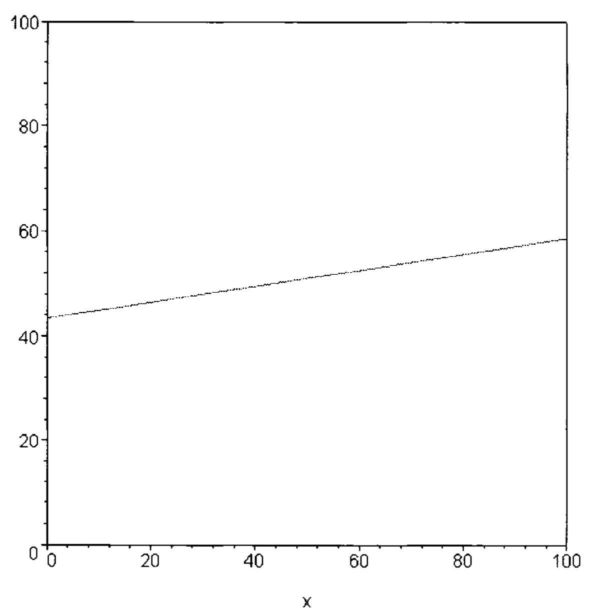

Figure 4.15: Allocation decision with the starting solution $A_1 = 10$ and $A_2 = 50$ (Example 4.3).

Iteration(j)	A1	A2	A3	A4	Slope	Intercept	Type	M1'	M2'	F_1	F_2
1	10.000	50.000	45	50	0.125	43.750	III	100.82	106.49	0.8197	6.4907
2	9.809	43.713	45	50	0.147	43.553	III	99.92	100.39	-0.0823	0.3922
3	10.260	43.320	45	50	0.151	43.448	III	100.00	100.00	-0.0034	0.0018
4	10.263	43.326	45	50	0.151	43.448	III	100.00	100.01	-0.0014	0.0068
5	10.271	43.319	45	50	0.151	43.446	III	100.00	100.00	0.0000	0.0000

Table 4.3. Constructive solution algorithm iteration results for initial triggers $A_1 = 10$ and $A_2 = 50$ with L_2^2 (Example 4.3).

Example 4.4: Constructive Solution Approach- Euclidean-metric (L_2) Case

For consistency we use the same example as in the case of Squared Euclidean-metric. Hence our market region is a square , $\mathcal{M}=\{(x, y)|x \in (0, 100)$, and $y \in (0, 100)\}$, i.e. $M = 100$ and demand density function is linear ($D(x, y) = 100 + 10x + 5y$) over the market region $\mathcal{M}$. The starting solution for this instance is $A1$=35 and $A2$ =40 at $y_{P1} = A3 = 40$ and $y_{P2} = A4 = 50$, respectively.

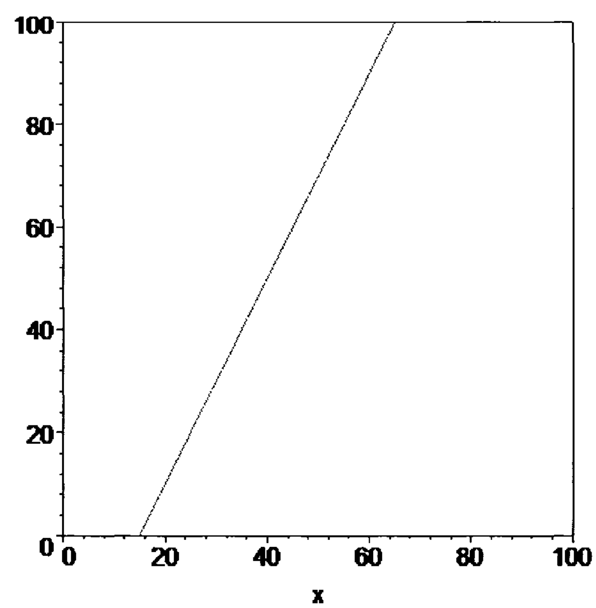

Figure 4.16: Allocation decision at the start of iteration 1 (Example 4.4).

The steps of the algorithm is as follows:

ITERATION 1

Step 1. Initialize the model parameters and variables

Set: $j = 1$

$y_{P1} = A3 = 40$ and $y_{P2} = A4 = 50$

$A1^{j=1}$=35 and $A2^{j=1}$ =40

$\epsilon_{BOUND} = 0.01$

$M = 100$

Step 2. Update the first service region sizes $A1^{j=1}$ and $A2^{j=1}$.

Step 2.1. Calculate Slope(a) and Intercept(b) of BR using $\left(\frac{y-A3}{A3-A4}\right) = \left(\frac{x-A1^{j=1}}{A1^{j=1}-A2^{j=1}}\right)$

Slope $a := 2$, Intercept $b := -30$

$br(x) = y = 2x - 30$, $br(y)^{-1} = x = \frac{y}{2} + 15$

$Y_{BR} = [0, 100]$, $X_{BR} = [15, 65]$, $X_{A1} = [15, 0]$, $X_{A2} = [65, 100]$

Step 2.2. Parametrize $\mathcal{A}_1$ and $\mathcal{A}_2$ in terms of $(M')^{j=1}$

Type I: If $a \geq 0$ and $br(x = 100) = 170 \geq M$ =100 and $br(x = 0) = -30 \leq 0$ then

$$\mathcal{A}_1 := \{(x, y)|y \in Y_{BR} = [0, 100] \text{ and } x \in [0, \tfrac{y}{2} + 15]\}$$

$$\mathcal{A}_2 := \{(x, y)|y \in Y_{BR} = [0, 100] \text{ and } x \in [\tfrac{y}{2} + 15, (M')^{j=1}]\}$$

Step 2.3. Solve the following single facility location problems in terms of $(M')^{j=1}$

For the first area, $\mathcal{A}_1$, we start with the initial estimates using centroidal locations and employ Weiszfeld method to find the correct median locations. Hence we start with $(x_1^G, y_1^G) = (27.7197, 69.0713)$ and after **10 iterations of Weiszfeld method**, we obtain optimal median locations, $(x_1^*, y_1^*) = (28.7537, 71.8961)$.

For the second area's, $\mathcal{A}_2$, optimum locations, we cannot employ the Weiszfeld method with the unknown $(M')^{j=1}$ in the equation. Hence we tentatively locate them at the centroidal locations which can be expressed as a closed form function of $(M')^{j=1}$ as below.

$$x_2^G = \tfrac{2(M')^{j=1}}{3} - 11\tfrac{2}{3} - \tfrac{90.7326}{(M')^{j=1} - 44.0569} + \tfrac{4257.3999}{(M')^{j=1} + 114.0569}$$

$$y_2^G = 50 - \tfrac{373.9028}{(M')^{j=1} - 44.0569} + \tfrac{1207.2360}{(M')^{j=1} + 114.0569}$$

Step 2.4. Solve following for the boundary value $(M1')^{j=1}$

Now we start with an initial value of $(M1')^{j=1} = M = 100$ and solve for the (x_2^G, y_2^G) and then employ **Weiszfeld method** to find the correct median locations.

Next we perform a line search over $(M1')^{j=1}$ to find the value of $(M1')^{j=1}$ satisfying below equation.

$$[(x_1^* - 35)^2 - (x_2^* - 35)^2] - [(y_2^* - 40)^2 - (y_1^* - 40)^2] = 0$$

That value is $(M1')^{j=1}$ =87.732 with $(x_2^*, y_2^*) = (66.5122, 47.9598)$ as the median locations

Step 2.5. Solve following for the boundary value $(M2')^{j=1}$

Again we start with an initial value of $(M2')^{j=1} = M = 100$ and solve for the (x_2^G, y_2^G) and then employ **Weiszfeld method** to find the correct median locations.

Next we perform a line search over $(M2')^{j=1}$ to find the value of $(M2')^{j=1}$ satisfying below equation.

$[(x_1^* - 40)^2 - (x_2^* - 40)^2] - [(y_2^* - 50)^2 - (y_1^* - 50)^2] = 0$

That value is $(M2')^{j=1}$ =84.495 with $(x_2^*, y_2^*) = (64.44497, 47.1388)$ as the median locations

Step 2.6. Calculate F_1 and F_2

F_1 =87.732-100=-12.268

F_2 =84.495-100=-15.505

Step 2.7. (In reference to the formal representation of the algorithm, we skip this step since we use first order based updating for L_2)

Step 2.8. Assign

$$\begin{pmatrix} A1 \\ A2 \end{pmatrix}^{j=2} = \begin{pmatrix} 35 \\ 40 \end{pmatrix}^{j=1} - \lambda \begin{pmatrix} -12.268 \\ -15.505 \end{pmatrix} = \begin{pmatrix} 47.268 \\ 55.505 \end{pmatrix}$$

where λ is constant step size and assigned to be 1.

ITERATION 2

Step 1. Initialize the model parameters and variables

Set: $j = 2$

$y_{P1} = A3 = 40$ and $y_{P2} = A4 = 50$

$A1^{j=2}$=47.268 and $A2^{j=2}$ =55.505

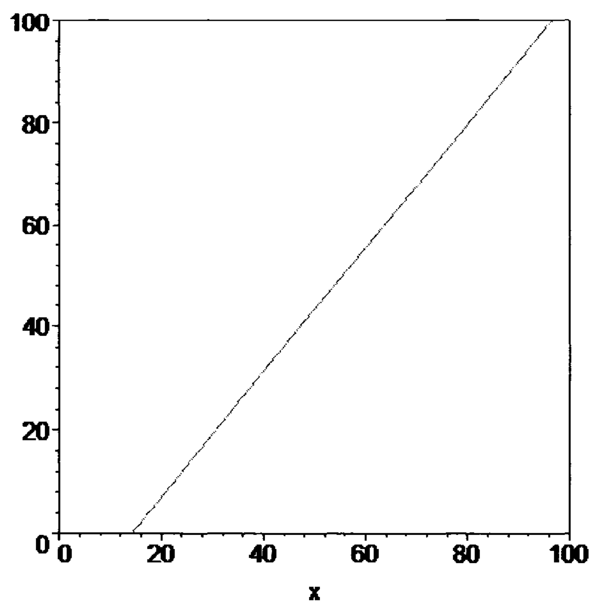

Figure 4.17: Allocation decision at the start of iteration 2 (Example 4.4).

$\epsilon_{BOUND} = 0.01$

$M = 100$

Step 2. Update the first service region sizes $A1^{j=2}$ and $A2^{j=2}$.

Step 2.1. Calculate Slope(a) and Intercept(b) of BR using $\left(\frac{y-A3}{A3-A4}\right) = \left(\frac{x-A1^{j=2}}{A1^{j=2}-A2^{j=2}}\right)$

Slope $a := 1.214$, Intercept $b := -17.385$

$br(x) = y = 1.214x - 17.385$, $br(y)^{-1} = x = .8237y + 14.32$

$Y_{BR} = [0, 100]$, $X_{BR} = [14.32, 96.69]$, $X_{A1} = [0, 14.32]$, $X_{A2} = [96.69, 100]$

Step 2.2. Parametrize $\mathcal{A}_1$ and $\mathcal{A}_2$ in terms of $(M')^{j=2}$

Type I: If $a \geq 0$ and $br(x = 100) = 104.02 \geq M = 100$ and $br(x = 0) = -17.385 \leq 0$ then

$\mathcal{A}_1 := \{(x, y)|y \in Y_{BR} = [0, 100] \text{ and } x \in [0, .8237y + 14.32]\}$

$\mathcal{A}_2 := \{(x, y)|y \in Y_{BR} = [0, 100] \text{ and } x \in [.8237y+14.32, (M')^{j=2}]\}$

Step 2.3. Solve the following single facility location problems in terms of

$(M')^{j=2}$

For $\mathcal{A}_1$, we again start with the initial estimates using centroidal locations and then employ the Weiszfeld's method to find the correct median locations. Hence, we start with $(x_1^G, y_1^G) = (41.6748, 70.7475)$, and after **6 iterations of Weiszfeld method**, we obtain optimal median locations, i.e. $(x_1^*, y_1^*) = (42.9261, 72.6472)$.

For the second area's, $\mathcal{A}_2$, optimal (median) locations, we cannot employ the Weiszfeld method with the unknown $(M')^{j=2}$ in the equation. Hence we tentatively locate them at the centroidal locations which can be expressed as a closed form function of $(M')^{j=2}$ as below.

$$x_2^G = \frac{2(M')^{j=2}}{3} - 11\tfrac{2}{3} - \frac{241.56}{(M')^{j=2} - 62.175} + \frac{6536876}{(M')^{j=2} + 132.175}$$

$$y_2^G = 50 - \frac{610.7025}{(M')^{j=2} - 62.174961} + \frac{1444.0358}{(M')^{j=2} + 132.17496}$$

Step 2.4. Solve following for the boundary value $(M1')^{j=2}$

Now we start with an initial value of $(M1')^{j=2} = M = 100$ and solve for the (x_2^G, y_2^G) and then employ **Weiszfeld's method** to find the correct median locations.

Next we perform a line search over $(M1')^{j=2}$ to find the value of $(M1')^{j=2}$ satisfying below equation.

$$[(x_1^* - 47.268)^2 - (x_2^* - 47.268)^2] - [(y_2^* - 40)^2 - (y_1^* - 40)^2] = 0$$

That value is $(M1')^{j=2}$ =103.40 with $(x_2^*, y_2^*) = (80.182, 41.183)$ as the median locations

Step 2.5. Solve following for the boundary value $(M2')^{j=2}$

Again we start with an initial value of $(M2')^{j=2} = M = 100$ and solve for the (x_2^G, y_2^G) and then employ **Weiszfeld method** to find the correct median locations.

Next we perform a line search over $(M2')^{j=2}$ to find the value of

$(M2')^{j=2}$ satisfying below equation.

$$[(x_1^* - 55.505)^2 - (x_2^* - 55.505)^2] - [(y_2^* - 50)^2 - (y_1^* - 50)^2] = 0$$

That value is $(M2')^{j=2}$ =102.86 with $(x_2^*, y_2^*) = (79.797, 40.986)$ as the median locations

Step 2.6. Calculate F_1 and F_2

F_1 =103.40-100=3.40

F_2 =102.86-100=2.86

Step 2.7. (In reference to the formal representation of the algorithm, we skip this step since we use first order based updating for L_2)

Step 2.8. Assign

$$\begin{pmatrix} A1 \\ A2 \end{pmatrix}^{j=3} = \begin{pmatrix} 47.268 \\ 55.505 \end{pmatrix}^{j=2} - \lambda \begin{pmatrix} 3.40 \\ 2.86 \end{pmatrix} = \begin{pmatrix} 43.868 \\ 52.645 \end{pmatrix}$$

where λ is the constant step size and assigned to be 1.

Note that since we are able to exactly determine the median locations of the $\mathcal{A}_1$ through Weiszfeld's procedure, $\mathcal{A}_2$'s optimal locations are consequently determined by performing a line search over the M', with a Weiszfeld iteration at each step. This is possible when the initial triggers $A1$ and $A2$ leads to allocation line alternatives **I, II, V**, and **VI** illustrated in Figure 4.9. In all other alternatives, $\mathcal{A}_1$ and $\mathcal{A}_2$ would both depend on M' and thus their optimal locations would depend on M'. In such a case, previously followed implementation of the shooting algorithm with integrated Weiszfeld procedure steps would be difficult. To overcome this difficulty, we **switch our initial trigger parameters** from being ($A1$,$A2$) to ($A3$,$A4$) and update ($A3$,$A4$) for given locations at ($A1$,$A2$). This is equivalent to **changing the reference**

axis from y-axis to x-axis, i.e. triggers are now chosen according to single-dimensional allocation decisions shown on the right in Figure 4.6. In order to illustrate this transition, we jump to **iteration 5** and detail this switch. Note that in step 2.2. of iteration 5, we change the service regions' notation; $\mathcal{A}_1$ is the service region below the allocation line and $\mathcal{A}_2$ is the one above. The boundary condition M' is now the **ceiling** of the square market region. With this change in notation, *iteration 5* could be executed as follows.

ITERATION 5[7]

Step 1. Initialize the model parameters and variables

Set: $j = 5$

$x_{P1} = A1^{j=5}$=40.048 and $x_{P2} = A2^{j=5}$ =52.715

$A3^{j=5} = 40$ and $A4^{j=5} = 50$

$\epsilon_{BOUND} = 0.01$

$M = 100$

Step 2. Update the first service region sizes $A3^{j=5}$ and $A4^{j=5}$.

Step 2.1. Calculate Slope(a) and Intercept(b) of BR using $\left(\frac{y-A3^{j=5}}{A3^{j=5}-A4^{j=5}}\right) = \left(\frac{x-A1}{A1-A2}\right)$

Slope $a :=$ .78945, Intercept b :=8.384

$br(x) = y = 0.78945x + 8.384$, $br(y)^{-1} = x = 1.2667y - 10.62$

$Y_{BR} = [8.384, 87.33]$, $X_{BR} = [0, 100]$, $Y_{A1} = [87.33, 100]$, $Y_{A2} = [0, 8.384]$

Step 2.2. Parametrize $\mathcal{A}_1$ and $\mathcal{A}_2$ in terms of $(M')^{j=5}$

Type III: If $a \geq 0$ and $br(x = 100) = 87.33 \leq M$ =100 and $br(x = 0) = 8.384 \geq 0$ then

$\mathcal{A}_1 := \{(x, y) | y \in [0, br(x) = 0.78945x + 8.384]$ and $x \in X_{BR}\}$

[7]Starting solution obtained from Iteration 4's results.

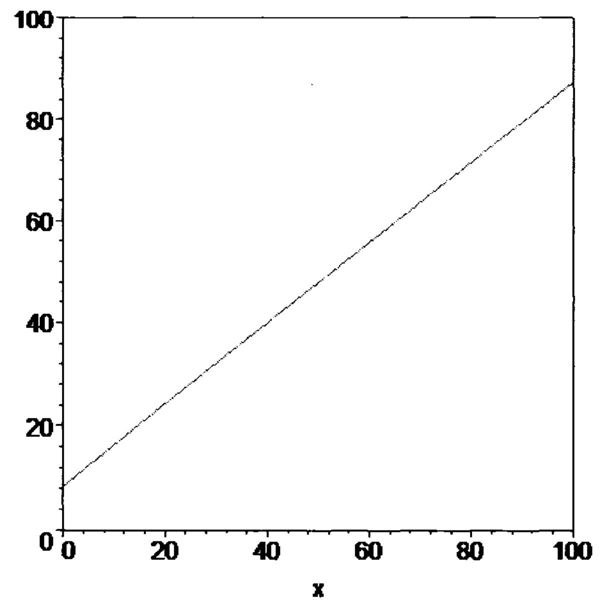

Figure 4.18: Allocation decision at the start of iteration 5 (Example 4.4).

$\mathcal{A}_2 := \{(x,y) | y \in [br(x) = 0.78945x + 8.384, (M')^{j=5}] \text{ and } x \in X_{BR}\}$

Step 2.3. Solve the following single facility location problems in terms of $(M')^{j=5}$

For $\mathcal{A}_1$, we start with the initial estimates using centroidal locations and employ Weiszfeld's method to find the correct median locations. Hence we start with $(x_1^G, y_1^G) = (72.4722, 34.6121)$ and after **8 iterations of Weiszfeld's method**, we obtain optimal (median) locations, $(x_1^*, y_1^*) = (74.35378, 34.96516)$.

For the second area's, $\mathcal{A}_1$, median locations, we cannot employ the Weiszfeld method with the unknown $(M')^{j=5}$ in the equation. Hence we tentatively locate them at the centroidal locations which can be expressed as a closed form function of $(M')^{j=5}$ as below.

$$x_2^G = -306679112.2 + 3833333.333\,(M')^{j=5} + 12500.0\left((M')^{j=5}\right)^2$$

$$y_2^G = \frac{2(M')^{j=5}}{3} - 40.0 + \frac{21089.7346}{(M')^{j=5} + 296.9936} - \frac{205.2526}{(M')^{j=5} - 56.9936}$$

Step 2.4. Solve following for the boundary value $(M1')^{j=5}$

Now we start with an initial value of $(M1')^{j=5} = M = 100$ and solve for the (x_2^G, y_2^G) and then employ **Weiszfeld's method** to find the correct median locations.

Next, we perform a line search over $(M1')^{j=5}$ to find the value of $(M1')^{j=5}$ satisfying below equation.

$[(x_1^* - 40.048)^2 - (x_2^* - 40.048)^2] - [(y_2^* - 40)^2 - (y_1^* - 40)^2] = 0$

That value is $(M1')^{j=5}$ =96.65 with $(x_2^*, y_2^*) = (47.356, 73.904)$ as the median locations.

Step 2.5. Solve following for the boundary value $(M2')^{j=5}$

Again we start with an initial value of $(M2')^{j=5} = M = 100$ and solve for the (x_2^G, y_2^G) and then employ **Weiszfeld's method** to find the correct median locations.

Next, we perform a line search over $(M2')^{j=5}$ to find the value of $(M2')^{j=5}$ satisfying below equation.

$[(x_1^* - 52.715)^2 - (x_2^* - 52.715)^2] - [(y_2^* - 50)^2 - (y_1^* - 50)^2] = 0$

That value is $(M2')^{j=5}$ =99.72 with $(x_2^*, y_2^*) = (48.394, 75.985)$ as the median locations.

Step 2.6. Calculate F_1 and F_2

F_1 =96.65-100=-3.35

F_2 =99.72-100=-0.28

Step 2.7. (In reference to the formal representation of the algorithm, we skip this step since we use first order based updating for L_2)

Step 2.8. Assign

$$\begin{pmatrix} A3 \\ A4 \end{pmatrix}^{j=6} = \begin{pmatrix} 40 \\ 50 \end{pmatrix}^{j=5} - \lambda \begin{pmatrix} -3.35 \\ -0.28 \end{pmatrix} = \begin{pmatrix} 43.35 \\ 50.28 \end{pmatrix}$$

where λ is the invariant step size and assigned to be 1.

When we continue until **iteration 9**, we would obtain the following local optimum solution.

$$\begin{pmatrix} A1 \\ A2 \end{pmatrix}^{j=9} = \begin{pmatrix} 40.048 \\ 52.715 \end{pmatrix}$$

$$and$$

$$\begin{pmatrix} A3 \\ A4 \end{pmatrix}^{j=9} = \begin{pmatrix} 48.72 \\ 51.00 \end{pmatrix}$$

$$\begin{aligned} \mathbf{x}_1^* &= (x_1^*, y_1^*) = (67.39, 28.43) \\ \mathbf{x}_2^* &= (x_2^*, y_2^*) = (58.63, 77.26) \end{aligned}$$

$$\begin{aligned} br(x) &: \ y = 0.18x + 41.512 \\ TC &= 236,344,838.9 \end{aligned}$$

Note that at iteration 5, we keep x-dimensional allocation decisions $A1$ and $A2$ constant and change the y-dimensional allocation decisions $A3$ and $A4$. Accordingly, the boundary value compared has become the ceiling of the square market region rather than the right-hand edge. Complete results are displayed in Table 4.4. Resulting allocation decision is illustrated in Figure 4.19.

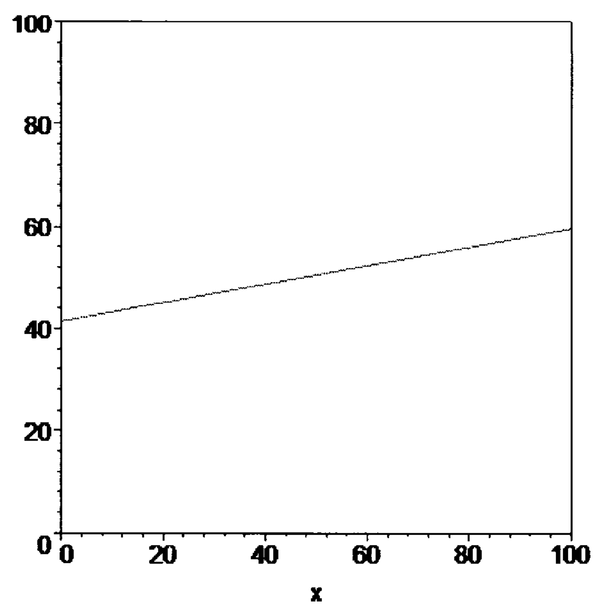

Figure 4.19: Allocation decision in the end of iteration 9 (Example 4.4).

Iteration(j)	A1	A2	A3	A4	Slope	Intercept	Type	M1'	M2'	F_1	F_2
1	35.000	40.000	40	50	2.000	-30.000	I	87.73	84.50	-12.268	-15.505
2	47.268	55.505	40	50	1.214	-17.385	I	103.40	102.86	3.400	2.860
3	43.868	52.645	40	50	1.139	-9.981	I	101.19	99.70	1.190	-0.300
4	42.678	52.945	40	50	0.974	-1.568	IV	102.63	100.23	2.630	0.230
5	40.048	52.715	40	50	0.789	8.384	III	96.65	99.72	-3.350	-0.280
6	40.048	52.715	43.35	50.28	0.547	21.440	III	96.88	99.79	-3.120	-0.210
7	40.048	52.715	46.47	50.49	0.317	33.760	III	98.11	99.64	-1.890	-0.360
8	40.048	52.715	48.36	50.85	0.197	40.488	III	99.60	99.84	-0.400	-0.160
9	40.048	52.715	48.76	51.01	0.178	41.646	III	100.04	100.01	0.040	0.010
10	40.048	52.715	48.72	51.00	0.180	41.512	III	100.01	100.00	0.010	0.000

Table 4.4. Constructive solution algorithm iteration results for initial triggers $A_1 = 35$ and $A_2 = 40$ with L_2 (Example 4.4).

4.4.2 Improvement Based- Steepest Descent Method

In this subsection, we first present the concept of shape-preserving transformations for both of the Euclidean-metric based distance measures. This concept allows us to design two improvement-based solution procedures, namely steepest-descent and modified Newton's methods. In what follows, we explain this shape-preserving concept and then illustrate its contribution in designing solution methods by presenting the steepest-descent algorithm for both distance measures. Modified Newton's method is developed in the last section.

Shape Preserving Transformations

The purpose of this section is to develop an approach for improving the allocation decisions of the location-allocation problem in allocation variable space (LAM-AVS). This improvement approach forms the basis for both the first-order steepest-descent improvement algorithm in this section as well as the second-order method presented in the next section. In comparison with improving location decisions of LAM-LVS, improving allocation decisions of LAM-AVS is more complicated. To improve locational decisions, one needs to change two coordinate values, x and y, for a given facility. On the other hand, allocation decisions are sets of an infinite number of points thus requires a different type of transformation than a single-point iteration. Before explaining this transformation, we first present the LAM-AVS in terms of single-dimensional allocation decisions, i.e. $A^y(x)$ and $A^x(y)$. More specifically, we will explicitly express the objective function in terms of these variables. Then we will derive the relationships between these allocation decisions and the slope and intercept of the allocation line BR. For any distance measure metric p, the objective function is as follows:

$$TC(A^y(x), A^x(y), \mathbf{x}_1^*, \mathbf{x}_2^*) = \int_{\mathcal{A}_1} d_p(\mathbf{x}_1^*, \mathbf{x}) D(\mathbf{x}) d\mathbf{x} + \int_{\mathcal{A}_2} d_p(\mathbf{x}_2^*, \mathbf{x}) D(\mathbf{x}) d\mathbf{x} \quad (55)$$

For brevity, we will represent $TC(A^y(x), A^x(y), \mathbf{x}_1^*, \mathbf{x}_2^*)$ with a shorter notation TC. It is possible to define TC in terms either $A^y(x)$ or $A^x(y)$. In Figure 4.20,

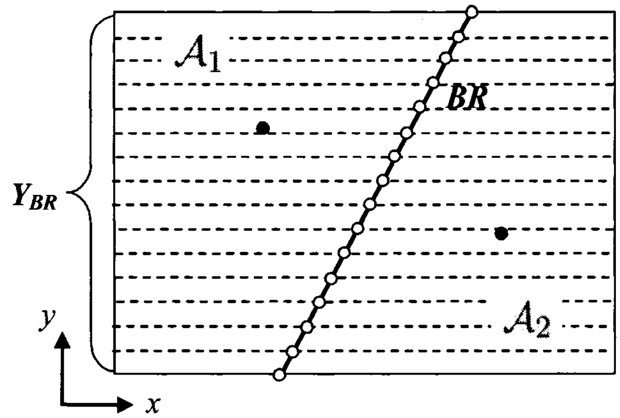

Figure 4.20: Discretization of the problem using grids which are parallel to x-axis and defined for each y.

we take an infinite number of strips on the y-axis and define the BR using the function $A^x(y)$. This formulation measures all of the two dimensional travel (x and y dimensions) using only the horizontal strips.

$$TC = \int_{y \in Y_{BR}} \int_0^{A^x(y)} d_p(\mathbf{x}_1^*, \mathbf{x}) D(\mathbf{x}) d\mathbf{x} + \int_{y \in Y_{BR}} \int_{A^x(y)}^{M} d_p(\mathbf{x}_2^*, \mathbf{x}) D(\mathbf{x}) d\mathbf{x} \quad (56)$$

Alternatively, we could take vertical strips and define the BR using the function $A^y(x)$ as in the Figure 4.21. This formulation measures all of the two-dimensional travel (in x- and y-dimensions) using only the vertical strips.

$$\begin{aligned} TC \;&=\; \int_{x \in X_{A1}} \int_0^{M} d_p(\mathbf{x}_1^*, \mathbf{x}) D(\mathbf{x}) d\mathbf{x} + \int_{x \in X_{BR}} \int_{A^y(x)}^{M} d_p(\mathbf{x}_1^*, \mathbf{x}) D(\mathbf{x}) d\mathbf{x} \quad (57) \\ &+ \int_{x \in X_{BR}} \int_0^{A^y(x)} d_p(\mathbf{x}_2^*, \mathbf{x}) D(\mathbf{x}) d\mathbf{x} + \int_{x \in X_{A2}} \int_0^{M} d_p(\mathbf{x}_2^*, \mathbf{x}) D(\mathbf{x}) d\mathbf{x} \end{aligned}$$

Recall from the model formulations (LAM, LAM-LVS, LAM-AVS) that we have the joint constraint (58), which ensures a feasible allocation decision. It stipulates that single-dimensional allocation decisions must map to the same

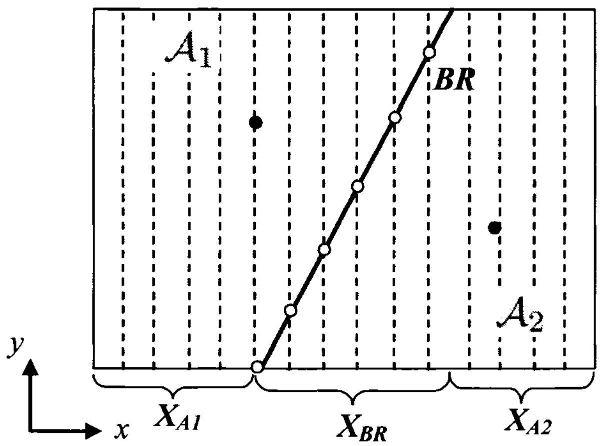

Figure 4.21: Discretization of the problem using grids which are parallel to y-axis and defined for each x.

point on the allocation line BR, i.e. they must form a line in Euclidean-metric based measures. When we move from a feasible solution of allocation line BR, i.e. satisfying below constraint, to another feasible BR, we must follow a **shape preserving transformation** on the plane.

$$(A^x(y), y) = (x, A^y(x)) \qquad for\ x \in X_{BR} \text{ and } y \in Y_{BR} \tag{58}$$

When BR is a straight line, as in the case of Euclidean-metric and Squared Euclidean-metric, there are two types of such transformation: **Rotation** and **Translation**. Herein, we will define the term "**Pure Rotation**" where the line rotates around $(0, b)$, namely only its slope (a) changes. In the other form of rotation around an axis of rotation, i.e. reference point which is fixed during rotation, both the slope(a) and intercept(b) changes. In the case of translation, only the intercept changes. Since by changing only the slope or the intercept, we cannot access all feasible solutions in the allocation decision space, we need to change them at the same time or alternate between pure

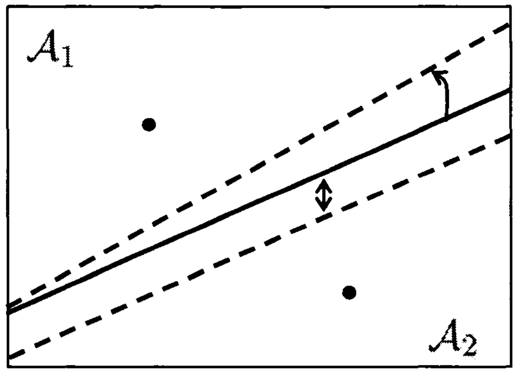

Figure 4.22: Shape preserving transformation for Euclidean-metric based allocation decisions.

rotation and translation. In fact, changing both the slope and the intercept is equivalent to rotating the allocation line BR around a reference point, which we will discuss in more detail shortly.

Figure 4.22 illustrates shape preserving transformation for Euclidean-metric based distance measures. In brief, Euclidean-metric based allocation line requires both rotation and translation to access all admissible solutions. Since the case for Euclidean-metric based shape preserving transformations include rotation in addition to the simple translation, there is need for some additional analysis. In the rest of this section, we will propose some results which lay the groundwork of the improvement algorithms for Euclidean-metric based problems.

Let's define **the reference point** around which we rotate the allocation line:

(x_r, y_r) : a point which is the axis of rotation for BR

Proposition 4.3. *For Euclidean-metric based distance measure cases, the derivative of the single dimensional allocation decisions ($A^y(x)$ and $A^x(y)$) with respect to the slope (a) of BR when rotated around a reference point*

(x_r, y_r) *is as follows:*

$$\frac{dA^x(y)}{da_r} = \frac{(y_r - y)}{a^2}$$

$$\frac{dA^y(x)}{da_r} = (x - x_r)$$

Proof.

Proof can be found in Appendix 4. ■

From this proposition, it follows that when we rotate BR around the y axis (i.e. $y_r = b$), it is equivalent to changing only the slope a, i.e. **pure rotation**. This rotation point is $(x_r, y_r) = (0, b)$.

$$\begin{aligned}
\frac{dA^x(y)}{da} &= \frac{(y_r - y)}{a^2} = \frac{(b - (ax + b))}{a^2} \\
&= -\frac{x}{a} = -\frac{A^x(y)}{a} \\
\frac{dA^y(x)}{da} &= (x - x_r) = x = A^x(y)
\end{aligned}$$

Similarly, as $y_r \to \infty$, $\frac{dA^x(y)}{da_r} = \frac{(y_r - y)}{a^2}$ becomes invariant to y, namely all the single dimensional decisions, $A^x(y)$, will change equal amount. This is in a sense equivalent to translation where y coordinate does not affect amount of change in $A^x(y)$ and this change is $\frac{-1}{a}$. Therefore, we have the following relations for the translation, i.e. changing the intercept of BR.

$$\frac{dA^x(y)}{db} = \frac{-1}{a}$$

$$\frac{dA^y(x)}{db} = 1$$

Proposition 4.4. *For Euclidean-metric based distance measure, when we rotate the allocation line BR around a reference point, the following relationship holds true irrespective of the reference point.*

$$\begin{aligned}\frac{dA^y(x)}{da} &= (-a)\frac{dA^x(y)}{da}\\ \frac{dA^y(x)}{db} &= (-a)\frac{dA^x(y)}{db}\end{aligned}$$

Proof.

Proof can be found in Appendix 4. ■

Furthermore, the following proposition establishes the equivalence of "*pure rotation combined with a translation*" to "*rotation around a reference point* (x_r, y_r)".

Proposition 4.5. *Rotation around a reference point (x_r, y_r) is an equivalent transformation to first decreasing the intercept by x_r and then performing a pure rotation around $(0, b)$.*

Proof.

Proof can be found in Appendix 4. ■

So we could cover all the possible single dimensional solutions by a combination of a pure translation and pure rotation step which has an equivalent form of generalized rotation. Thus next corollary establishes similar differential equivalence for the objective function.

Corollary 4.1. *Rate of change in the objective function with the rotation around a reference point (x_r, y_r) is equivalent to rate of change as decreasing intercept by x_r and then performing a pure rotation around $(0, b)$.*

$$\frac{dTC}{da_r} = \frac{dTC}{da} - x_r\frac{dTC}{db}$$

where $\frac{dTC}{da_r}$ is the change in TC when we rotate BR around a reference point (x_r, y_r).

Note that we could represent all possible movements of the line, i.e. any combination of pure rotation and translation, by rotating around a reference point. In the next section, we will describe the first-order steepest descent method for both Euclidean-metric based distance measures.

Steepest-Descent Algorithm

In this section, we first present the gradient of objective function with respect to rotation around a reference point and then provide the formal improvement algorithms for both Euclidean-metric based distance measures (L_2 and L_2^2) with example applications. We denote the derivative of the objective function with respect to rotation around any reference point as $\frac{dTC}{da_r}$. Following proposition delineates this derivative.

Proposition 4.6. *The derivative of objective function with respect to the slope when it is rotated around a reference point (x_r, y_r) can be found through either (59) or (60). (59) and (60) corresponds to the horizontal and vertical representations of TC in (56) and (57), respectively.*

$$\frac{dTC}{da_r} = \int_{y \in Y_{BR}} \left(\frac{\partial TC}{\partial A^x(y)} - \frac{\partial TC}{\partial A^y(x)} \right) \frac{\partial A^x(y)}{\partial a_r} dy \tag{59}$$

$$\frac{dTC}{da_r} = \int_{x \in X_{BR}} \left(\frac{\partial TC}{\partial A^y(x)} - \frac{\partial TC}{\partial A^x(y)} \right) \frac{\partial A^y(x)}{\partial a_r} dx \tag{60}$$

Proof.

Proof can be found in Appendix 4. ■

Next proposition presents the functional form of the partial derivatives of TC with respect to single dimensional allocation decisions.

Proposition 4.7.

The partial derivatives of the objective function with respect to single-dimensional allocation decisions satisfy the following relationship.

$$\left(\frac{\partial TC}{\partial A^x(y)} - \frac{\partial TC}{\partial A^y(x)} \right) = \left[d_p(\mathbf{x}_1^*, \mathbf{x}) - d_p(\mathbf{x}_2^*, \mathbf{x}) \right] D(\mathbf{x}) \tag{61}$$

where $x = (A^x(y), A^y(x)) \in BR$ and for $p = L_2$ and $p = L_2^2$.

Proof.

Proof can be found in Appendix 4. ■

Note that Propositions 4.6. and 4.7. establish an analogous result to the first order condition obtained in the single dimensional chapter. Specifically, when the allocation line (BR) is such that every point on the line is at **equidistant** to the locations, then the derivative of the system cost with respect to slope and intercept, $\frac{dTC}{da}$ and $\frac{dTC}{db}$, is zero. This property is the first-order optimality condition for the LAM-AVS.

Note that Squared Euclidean-metric (L_2^2) is a separable distance measure. Accordingly the objective function for such types of separable distance measures can be written as a mixed measure of x- and y-dimensional travel using both the horizontal and vertical strips. This mixed measure is expressed as follows. Figure 4.23 further illustrates this mixed form of distance measure.

$$\begin{aligned} TC &= \int_{y \in Y_{BR}} \int_0^{A^x(y)} d_{p_x}(\mathbf{x}_1^*, \mathbf{x}) D(\mathbf{x}) d\mathbf{x} + \int_{y \in Y_{BR}} \int_{A^x(y)}^M d_{p_x}(\mathbf{x}_2^*, \mathbf{x}) D(\mathbf{x}) d\mathbf{x} \\ &+ \int_{x \in X_{A1}} \int_0^M d_{p_y}(\mathbf{x}_1^*, \mathbf{x}) D(\mathbf{x}) d\mathbf{x} + \int_{x \in X_{BR}} \int_{A^y(x)}^M d_{p_y}(\mathbf{x}_1^*, \mathbf{x}) D(\mathbf{x}) d\mathbf{x} \\ &+ \int_{x \in X_{BR}} \int_0^{A^y(x)} d_{p_y}(\mathbf{x}_2^*, \mathbf{x}) D(\mathbf{x}) d\mathbf{x} + \int_{x \in X_{A2}} \int_0^M d_{p_y}(\mathbf{x}_2^*, \mathbf{x}) D(\mathbf{x}) d\mathbf{x} \end{aligned} \tag{62}$$

where

$$d_p(\mathbf{x}_i^*, \mathbf{x}) = d_{p_x}(\mathbf{x}_i^*, \mathbf{x}) + d_{p_y}(\mathbf{x}_i^*, \mathbf{x})$$

Squared Euclidean-metric (L_2^2):

$$\begin{aligned} d_{p_x}(\mathbf{x}_i^*, \mathbf{x}) &= (x_i^* - x)^2 \\ d_{p_y}(\mathbf{x}_i^*, \mathbf{x}) &= (y_i^* - y)^2 \end{aligned}$$

Due to this separable property of Squared Euclidean-metric based distance measure, the partial derivative with respect to single-dimensional allocation decisions could be independently expressed. Following proposition illustrates this characteristic.

Proposition 4.8. *The partial derivative of the objective function with respect to single-dimensional allocation decisions satisfies the following relationship*

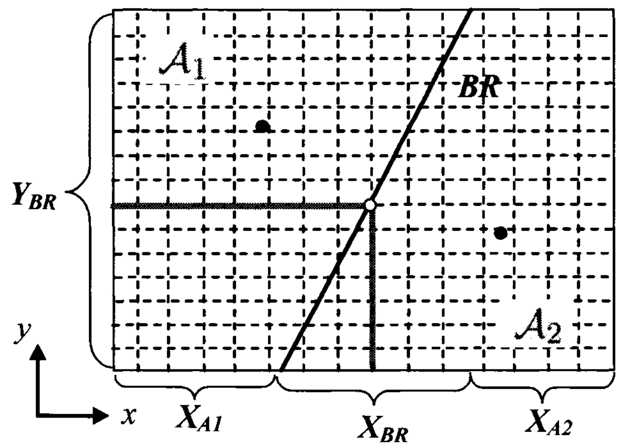

Figure 4.23: Discretization of the problem using grids parallel to x- annd y-coordinate axes.

when the distance measure is separable, i.e. $d_p(\mathbf{x}_i^*, \mathbf{x}) = d_{p_x}(\mathbf{x}_i^*, \mathbf{x}) + d_{p_y}(\mathbf{x}_i^*, \mathbf{x})$

$$\frac{\partial TC}{\partial A^x(y)} = [d_{p_x}(\mathbf{x}_1^*, \mathbf{x}) - d_{p_x}(\mathbf{x}_2^*, \mathbf{x})]\, D(\mathbf{x}) \tag{63}$$

$$\frac{\partial TC}{\partial A^y(x)} = \left[d_{p_y}(\mathbf{x}_1^*, \mathbf{x}) - d_{p_y}(\mathbf{x}_2^*, \mathbf{x})\right] D(\mathbf{x}) \tag{64}$$

where $x = (A^x(y), A^y(x)) \in BR$

Proof.

Proof can be found in Appendix 4. ∎

In what follows, we provide the steepest-descent algorithm for the L_2 and L_2^2, for which the allocation line BR is a straight line. Note that, so far, we have derived gradient properties of the total cost with respect to the slope (a) and intercept (b), so a natural implementation of steepest-descent algorithm would be to use a and b as iterates in the algorithm. However, for consistency, we will implement the algorithm in terms of two triggers as in the case of

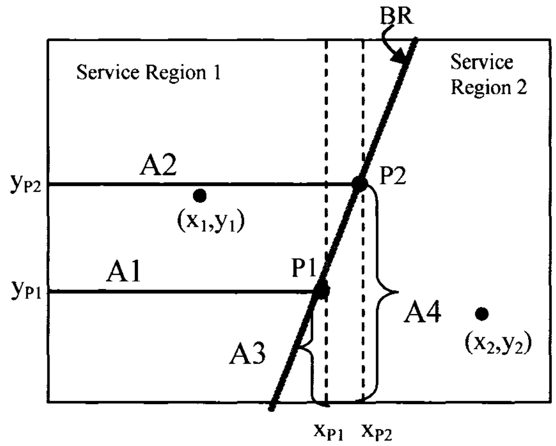

Figure 4.24: Steepest-descent improvement algorithm example for Euclidean-metric.

constructive algorithm. These triggers serve as a construct for implementation of the algorithm in a way that is consistent with the constructive algorithm. Let's describe the use of these triggers in relation to the slope and intercept iterations using Figure 4.24. We can express the slope and intercept of the allocation line BR using the pair $(A1, A3)$ and $(A2, A4)$ as follows.

$$a = \frac{A3 - A4}{A1 - A2} \quad \text{and} \quad b = A3 - A1\,(a)$$

Accordingly, $A1$ and $A2$ can be expressed as in (65), which is used in the algorithm to achieve the consistency with the constructive approach.

$$A1 = \frac{A3 - b}{a} \quad \text{and} \quad A2 = \frac{A4 - b}{a} \tag{65}$$

Steepest-Descent Improvement Algorithm:

Step 1. Define and Initialize the model parameters and variables

j : index for optimality iterations (i.e. $j^* = \{j | \ \epsilon_{COST} \geq \frac{|TC^{j+1} - TC^j|}{|TC^j|}\}$

ϵ_{COST} : epsilon parameter for optimality stopping decision

α^j : step length for line search iterations at the jth iteration

$A1^j$: j^{th} iteration value for the first service region size at $y = y_{P1}$

$A2^j$: j^{th} iteration value for the first service region size at $y = y_{P2}$

$\mathbf{x}_i^* = (x_i^*, y_i^*)$: optimal locations corresponding to $\mathcal{A}^j_{i=1,2}$

M : market boundary parameter

Set $j = 0$, $A1^{j=0}$ and $A2^{j=0}$

Step 2. Initialization: Allocate the service regions and optimally locate facilities

Step 2.1. Calculate Slope(a) and Intercept(b) of the initial BR using $\left(\frac{y-A3}{A3-A4}\right) = \left(\frac{x-A1^j}{A1^j-A2^j}\right)$

Step 2.2. Identify the following sets and functions given BR

$\mathcal{A}_1$, $\mathcal{A}_2$,

Step 2.3. Find the optimal locations and calculate the total cost

$$\mathbf{x}_1^* = (x_1^*, y_1^*)^j := \arg \min_{(x_1,y_1)} (\textstyle\int_{\mathcal{A}_1} d_p(\mathbf{x}_1, \mathbf{x}) D(\mathbf{x}) d\mathbf{x})$$
$$\mathbf{x}_2^* = (x_2^*, y_2^*)^j := \arg \min_{(x_2,y_2)} (\textstyle\int_{\mathcal{A}_2} d_p(\mathbf{x}_2, \mathbf{x}) D(\mathbf{x}) d\mathbf{x})$$
$$TC^j = \textstyle\int_{\mathcal{A}_1} d_p(\mathbf{x}_1^*, \mathbf{x}) D(\mathbf{x}) d\mathbf{x} + \int_{\mathcal{A}_2} d_p(\mathbf{x}_2^*, \mathbf{x}) D(\mathbf{x}) d\mathbf{x}$$

Step 3. Improvement: Update the first service region sizes $A1^j$ and $A2^j$

Do While $(\frac{|TC^{j+1} - TC^j|}{|TC^j|} \geq \epsilon_{COST})$:

$j = j + 1$

Step 3.1. Calculate the partial gradients

· Total cost with respect to single-dimensional allocation decisions $(A^x(y), A^y(x))$

$$\left(\frac{\partial TC}{\partial A^x(y)} - \frac{\partial TC}{\partial A^y(x)}\right) = \left[d_p(\mathbf{x}_1^*, \mathbf{x}) - d_p(\mathbf{x}_2^*, \mathbf{x})\right] D(\mathbf{x})$$

· Total cost with respect to:

Slope (Pure rotation): $\frac{dTC}{da} = \int_{y \in Y_{BR}} \left(\frac{\partial TC}{\partial A^x(y)} - \frac{\partial TC}{\partial A^y(x)} \right) \frac{\partial A^x(y)}{\partial a} dy$

Intercept (Translation): $\frac{dTC}{db} = \int_{y \in Y_{BR}} \left(\frac{\partial TC}{\partial A^x(y)} - \frac{\partial TC}{\partial A^y(x)} \right) \frac{\partial A^x(y)}{\partial b} dy$

where $\frac{\partial A^x(y)}{\partial a} = -\frac{A^x(y)}{a}$ and $\frac{\partial A^x(y)}{\partial b} = -\frac{1}{a}$

· Normalize the gradients

$$d^j_{SLOPE} = \frac{\frac{dTC}{da}}{\sqrt{\left(\frac{dTC}{da}\right)^2 + \left(\frac{dTC}{db}\right)^2}}$$
$$d^j_{INTERCEPT} = \frac{\frac{dTC}{db}}{\sqrt{\left(\frac{dTC}{da}\right)^2 + \left(\frac{dTC}{db}\right)^2}}$$

Step 3.2. Perform a line search for step size α^j

· Update the two allocation decisions $A1^j$ and $A2^j$

$$(A1^j)' = \frac{A3 - \left(b^j - \alpha^j d^j_{INTERCEPT}\right)}{\left(a^j - \alpha^j d^j_{SLOPE}\right)}$$
$$(A2^j)' = \frac{A4 - \left(b^j - \alpha^j d^j_{INTERCEPT}\right)}{\left(a^j - \alpha^j d^j_{SLOPE}\right)}$$

· Repeat **Steps 2.1, 2.2, 2.3** using $(A1^j)'$ and $(A2^j)'$

· Find $(\alpha^j)^* = \arg\min_{\alpha^j} TC$

Step 3.3. Update the allocation decisions

$$A1^{j+1} = \frac{A3 - \left(b^j - (\alpha^j)^* d^j_{INTERCEPT}\right)}{\left(a^j - (\alpha^j)^* d^j_{SLOPE}\right)}$$
$$A2^{j+1} = \frac{A4 - \left(b^j - (\alpha^j)^* d^j_{INTERCEPT}\right)}{\left(a^j - (\alpha^j)^* d^j_{SLOPE}\right)}$$

· Repeat **Steps 2.2, 2.3** using $A1^{j+1}$ and $A2^{j+1}$

· Return to **Step 3**

Step 4. Terminate with the solution $A1^j$, $A2^j$ and BR

The only difference between L_2 and L_2^2 is that in the L_2^2 case there are closed form solutions for the centroids whereas we need to resort to the Weiszfeld's method to numerically calculate the median locations for the L_2 case.

Example 4.5: Steepest-Descent Improvement Algorithm - Euclidean-metric (L_2) Case

Let's consider an example implementation of the above algorithm for the Figure 4.24. We will use the same example in the preceding sections. We have a square market region $\mathcal{M}$=$\{(x, y)|x \in (0, 100)$ and $y \in (0, 100)\}$, i.e. $M = 100$. We wish to determine an optimal allocation decision for a linear demand density function ($D(x, y) = 100+10x+5y$) over the market region $\mathcal{M}$. The starting solution for this instance is $A1$=35 and $A2 = 40$ at $y_{P1} = A3 = 40$ and $y_{P2} = A4 = 50$, respectively.

The steps of the algorithm are as follows:

ITERATION 1

Step 1. Define and Initialize the model parameters and variables

$j = 0$

$\epsilon_{COST} := 5 \times 10^{-4}$

$y_{P1} = A3 = 40$ and $y_{P2} = A4 = 50$

$A1^{j=1}$=35 and $A2^{j=1} = 40$

$M = 100$

Step 2. Initialization: Allocate the service regions and optimally locate facilities

Step 2.1. Calculate Slope(a) and Intercept(b) of BR using $\left(\frac{y-A3}{A3-A4}\right) = \left(\frac{x-A1^j}{A1^j-A2^j}\right)$

Slope $a := 2$, Intercept $b := -30$

Step 2.2. Identify the following sets and functions given BR

$br(x) = y = 2x + 30$, $br(y)^{-1} = x = \frac{y}{2} + 15$

$Y_{BR} = [0, 100]$, $X_{BR} = [15, 65]$, $X_{A1} = [0, 15]$, $X_{A2} = [65, 100]$

Step 2.3. Find the optimal locations and calculate the total cost

Start with the initial estimates using centroidal locations:

$(x_1^G, y_1^G) = (27.72, 69.07)$ and $(x_2^G, y_2^G) = (73.27, 48.96)$

Employ **Weiszfeld's method** to find the correct median locations.

$(x_1^*, y_1^*)^{j=1} := (28.72, 71.89)$

$(x_2^*, y_2^*)^{j=1} := (74.22, 49.94)$

$TC^{j=1} = 248,018,120.86$

Step 3. Improvement: Update the first service region sizes $A1^{j=1}$ and $A2^{j=1}$

Do While $(\frac{|TC^{j+1} - TC^j|}{|TC^j|} \geq \epsilon_{COST})$:

Step 3.1. Calculate the partial gradients

· Total cost with respect to Single dimensional allocation decisions $(A^x(y), A^y(x))$

Slope (Pure rotation): $\frac{dTC}{da}$ = 13,446,950.0

Intercept (Translation): $\frac{dTC}{db}$ = 292,916.5

Normalize the gradients

$d_{SLOPE}^{j=1} = 0.9998$

$d_{INTERCEPT}^{j=1} = 0.0218$

Step 3.2. Update the two allocation decisions $A1^{j=1}$ and $A2^{j=1}$

$(A1^{j=1})' = \frac{70 - 0.0218\alpha}{2 - 0.9998\alpha}$

$(A2^{j=1})' = \frac{80 - 0.0218\alpha}{2 - 0.9998\alpha}$

· Repeat **Steps 2.1, 2.2, 2.3** using $(A1^{j=1})'$ and $(A2^{j=1})'$

· $(\alpha^{j=1})^* = \arg\min_{\alpha^j} TC = 0.5660$

Step 3.3. Update the allocation decisions

$A1^{j=2} = \frac{40-(-30-(0.5660)0.0218)}{(2-(0.5660)0.9998)}$ =48.82

$A2^{j=2} = \frac{50-(-30-(0.5660)0.0218)}{(2-(0.5660)0.9998)}$ =55.79

$TC^{j=2} = 243,423,874.69$

Return to **Step 3**

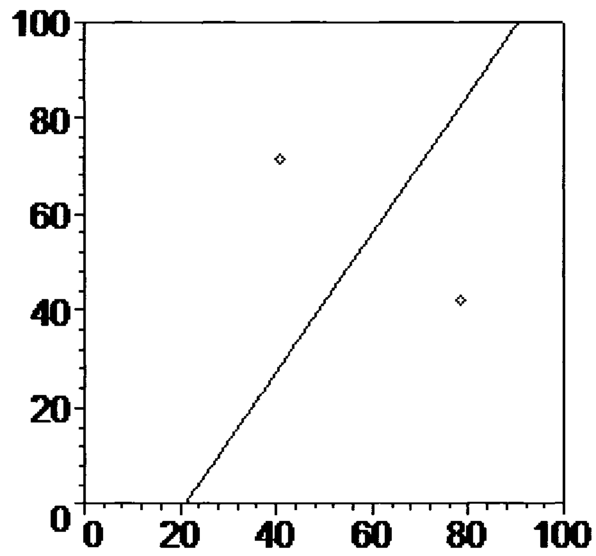

Figure 4.25: Solution in the end of first iteration of the Euclidean-metric based example for Steepest-descent improvement method (Example 4.5).

Allocation decisions in the end of iteration 1 is displayed in Figure 4.25. For brevity, we do not detail the remainder of iterations. Table 4.5. presents the results for the remaining iterations. Last column, **% GAP**, represents the percentage gap between the current iteration result and the solution of the constructive shooting approach. Optimality condition (less than 0.05% improvement) is reached at $j = 20^{th}$ iteration, which is illustrated in Figure 4.26. Note that **linear convergence rate** associated with the steepest-descent algorithm is apparent from the iteration results.

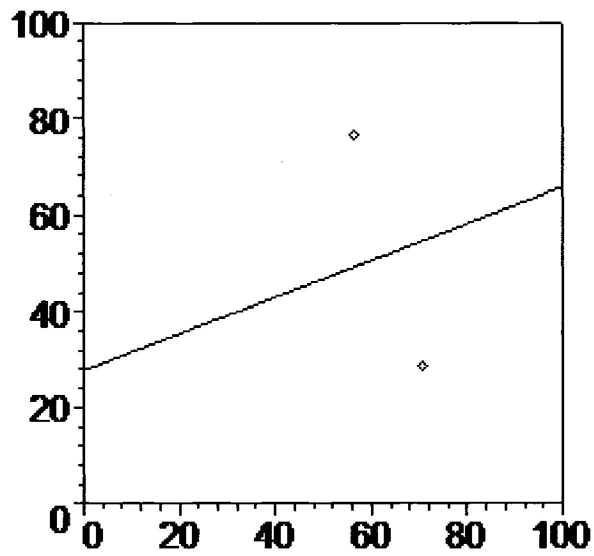

Figure 4.26: Solution in the end of 20^{th} iteration of the Euclidean-metric based example for Steepest-descent improvement method (Example 4.5).

		TRIGGERS				LOCATIONS				ALLOCATION LINE			
j	BR Type	A1	A2	A3	A4	x1	y1	x2	y2	Slope	Intercept	TC	% GAP
0	I	35.00	40.00	40	50	28.72	71.89	74.22	49.94	2.000	-30.000	248,018,074.90	4.98%
1	I	48.82	55.79	40	50	40.81	71.13	78.53	42.10	1.434	-30.012	243,423,874.69	3.04%
2	I	49.21	56.42	40	50	41.61	71.30	78.66	41.37	1.387	-28.263	243,368,196.10	3.01%
3	IV	53.27	61.34	40	50	45.91	71.36	79.14	37.09	1.239	-26.002	243,245,114.62	2.96%
4	IV	53.64	62.64	40	50	47.70	71.88	78.62	34.85	1.111	-19.582	242,575,784.83	2.68%
5	IV	53.27	63.01	40	50	48.88	72.31	78.08	33.70	1.027	-14.699	241,997,209.79	2.43%
6	IV	52.72	63.22	40	50	49.41	72.66	77.49	32.77	0.953	-10.226	241,429,767.00	2.19%
7	IV	51.93	63.21	40	50	50.39	73.08	76.80	32.05	0.887	-6.035	240,871,041.48	1.96%
8	IV	51.16	63.30	40	50	51.09	73.42	76.13	31.34	0.824	-2.130	240,335,087.17	1.73%
9	III	50.01	63.04	40	50	51.35	73.77	75.50	30.86	0.767	1.630	239,812,751.01	1.51%
10	III	48.87	62.87	40	50	51.94	74.10	75.02	30.41	0.714	5.088	239,335,498.54	1.31%
11	III	47.56	62.64	40	50	52.56	74.41	74.56	30.01	0.663	8.462	238,885,380.46	1.12%
12	III	47.65	64.23	40	50	53.63	74.35	74.09	29.01	0.603	11.264	238,556,442.79	0.98%
13	III	44.12	61.83	40	50	53.77	75.04	73.35	29.38	0.565	15.083	238,063,788.92	0.77%
14	III	42.70	61.58	40	50	54.25	75.24	72.87	29.15	0.530	17.380	237,803,843.05	0.66%
15	III	41.04	61.09	40	50	54.635	75.445	72.394	29.022	0.499	19.534	237,574,533.79	0.56%
16	III	39.38	60.67	40	50	55.033	75.624	71.948	28.891	0.470	21.505	237,378,626.51	0.48%
17	III	37.46	60.01	40	50	55.351	75.819	71.508	28.832	0.443	23.387	237,204,762.92	0.40%
18	III	35.67	59.54	40	50	55.693	75.971	71.118	28.742	0.419	25.054	237,062,372.58	0.34%
19	III	34.05	59.24	40	50	55.999	76.066	70.755	28.628	0.397	26.483	236,949,362.64	0.30%
20	III	32.08	58.48	40	50	56.222	76.224	70.441	28.641	0.379	27.843	236,850,228.50	0.25%

Table 4.5. Steepest-descent improvement algorithm's iteration results Euclidean-metric (L_2) example (Example 4.5).

4.4.3 Improvement Based: Modified-Newton Method

The purpose of this section is to illustrate performance gain by using second-order improvement methods over the gradient-based improvement method (i.e. steepest-descent) discussed in the previous section. Since the shape preserving transformation concept is developed in the previous section, we do not repeat here. Instead, we will first present an analytical form of the second-order derivatives for Euclidean-metric based cases, and then, provide the modified-Newton algorithm, which accounts for negative-definiteness of the Hessian, thus providing guaranteed convergence. Lastly, we will illustrate the performance difference between first-order and second-order methods on a simple example.

Due to the simplicity of allocation decision in 2-facility case, i.e. a straight line, we can calculate the second order derivatives without much difficulty. Following proposition provides the analytical form of the Hessian for the case when allocation line BR is a straight line, namely Euclidean-metric based distance measures $p = L_2$ and $p = L_2^2$.

Proposition 4.9. *Hessian of the TC with respect to the allocation line BR parametrized over its slope (a) and intercept (b), for the cases L_2 and L_2^2, can be found as follows:*

$$\nabla^2 TC = \begin{bmatrix} \frac{\partial^2 TC}{\partial a^2} & \frac{\partial^2 TC}{\partial a \partial b} \\ \frac{\partial^2 TC}{\partial a \partial b} & \frac{\partial^2 TC}{\partial b^2} \end{bmatrix}$$

where

$$\frac{d^2TC}{da^2} = \int_{y \in Y_{BR}} \left[\begin{array}{c} \left(\frac{\partial^2 TC}{\partial A^x(y)^2} + (a)\frac{\partial^2 TC}{\partial A^y(x)^2}\right)\left(\frac{\partial A^x(y)}{\partial a}\right)^2 \\ + \left(\frac{\partial TC}{\partial A^x(y)} - \frac{\partial TC}{\partial A^y(x)}\right)\frac{\partial^2 A^x(y)}{\partial a^2} \end{array} \right] dy$$

$$\frac{d^2TC}{db^2} = \int_{y \in Y_{BR}} \left(\frac{\partial^2 TC}{\partial A^x(y)^2} + (a)\frac{\partial^2 TC}{\partial A^y(x)^2}\right)\left(\frac{\partial A^x(y)}{\partial b}\right)^2 dy$$

$$\frac{\partial^2 TC}{\partial a \partial b} = \int_{y \in Y_{BR}} \left[\begin{array}{c} \left(\frac{\partial^2 TC}{\partial A^x(y)^2} + (a)\frac{\partial^2 TC}{\partial A^y(x)^2}\right)\frac{\partial A^x(y)}{\partial a}\frac{dA^x(y)}{db} \\ + \left(\frac{\partial TC}{\partial A^x(y)} - \frac{\partial TC}{\partial A^y(x)}\right)\frac{\partial^2 A^x(y)}{\partial a \partial b} \end{array} \right] dy$$

where $x = (A^x(y), A^y(x)) \in BR$.

$\frac{\partial TC}{\partial A^x(y)}, \frac{\partial TC}{\partial A^y(x)}, \frac{\partial^2 TC}{\partial A^x(y)^2}$ *and* $\frac{\partial^2 TC}{\partial A^y(x)^2}$ *can be obtained from (56) and (57). Moreover derivatives of* $A^x(y)$ *with respect to slope and intercept are as follows.*

$$\begin{aligned} \frac{\partial A^x(y)}{\partial a} &= -\frac{A^x(y)}{a} \text{ and } \frac{\partial^2 A^x(y)}{\partial a^2} = 2\frac{A^x(y)}{a^2} \\ \frac{\partial A^x(y)}{\partial b} &= -\frac{1}{a} \text{ and } \frac{\partial^2 A^x(y)}{\partial b^2} = 0 \\ \frac{\partial^2 A^x(y)}{\partial a \partial b} &= \frac{1}{a^2} \end{aligned}$$

Proof.

Proof can be found in Appendix 4. ■

We now present the second-order improvement algorithm. For consistency, we will implement the algorithm in terms of two triggers ($A1$ and $A2$) as in the case of constructive and steepest descent algorithm. Figure 4.27, repeated here for convenience, illustrates the notation used in the algorithm.

Modified-Newton Improvement Algorithm:

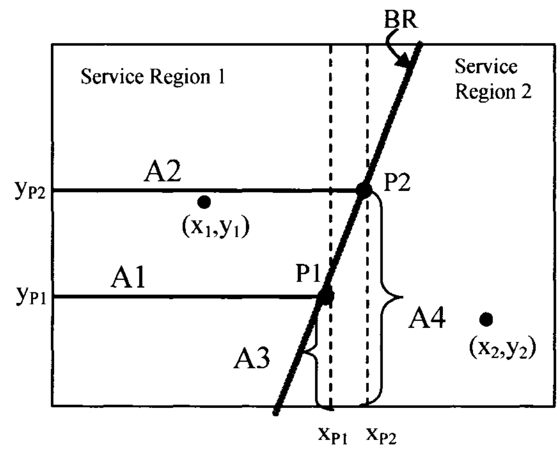

Figure 4.27: Illustration of the allocation decisions for 2-facility Euclidean-metric based distance measure.

Step 1. Define and Initialize the model parameters and variables

j : index for optimality iterations (i.e. $j^* = \{j| \ \epsilon_{COST} \geq \frac{|TC^{j+1}-TC^j|}{|TC^j|}\}$

ϵ_{COST} : epsilon parameter for optimality stopping decision

α^j : step length for line search iterations at the jth iteration

$A1^j$: j^{th} iteration value for the first service region size at $y = y_{P1}$

$A2^j$: j^{th} iteration value for the first service region size at $y = y_{P2}$

$\mathbf{x}_i^* = (x_i^*, y_i^*)$: optimal locations corresponding to $\mathcal{A}_{i=1,2}^j$

M : market boundary parameter

Set $j = 0$, $A1^j$ and $A2^j$

Step 2. Initialization: Allocate the service regions and optimally locate facilities

Step 2.1. Calculate Slope(a) and Intercept(b) of BR using $\left(\frac{y-A3}{A3-A4}\right) = \left(\frac{x-A1^j}{A1^j-A2^j}\right)$

Step 2.2. Identify the following sets and functions given BR

$\mathcal{A}_1$, $\mathcal{A}_2$,

Step 2.3. Find the optimal locations and calculate the total cost

$$\mathbf{x}_1^* = (x_1^*, y_1^*)^j := \arg \min_{(x_1,y_1)} (\int_{A_1} d_p(\mathbf{x}_1, \mathbf{x}) D(\mathbf{x}) d\mathbf{x})$$
$$\mathbf{x}_2^* = (x_2^*, y_2^*)^j := \arg \min_{(x_2,y_2)} (\int_{A_2} d_p(\mathbf{x}_2, \mathbf{x}) D(\mathbf{x}) d\mathbf{x})$$
$$TC^j = \int_{A_1} d_p(\mathbf{x}_1^*, \mathbf{x}) D(\mathbf{x}) d\mathbf{x} + \int_{A_2} d_p(\mathbf{x}_2^*, \mathbf{x}) D(\mathbf{x}) d\mathbf{x}$$

Step 3. Improvement: Update the first service region sizes $A1^j$ and $A2^j$

Do While $\left(\frac{|TC^{j+1} - TC^j|}{|TC^j|} \geq \epsilon_{COST}\right)$:

$j = j + 1$

Step 3.1. Calculate the Partial Derivatives and the Hessian

· Total cost with respect to single dimensional allocation decisions $(A^x(y), A^y(x))$

$$\left(\frac{\partial TC}{\partial A^x(y)} - \frac{\partial TC}{\partial A^y(x)}\right) = [d_p(\mathbf{x}_1^*, \mathbf{x}) - d_p(\mathbf{x}_2^*, \mathbf{x})]\, D(\mathbf{x})$$

· Total cost with respect to:

Slope (Pure rotation): $\frac{dTC}{da} = \int_{y \in Y_{BR}} \left(\frac{\partial TC}{\partial A^x(y)} - \frac{\partial TC}{\partial A^y(x)}\right) \frac{\partial A^x(y)}{\partial a} dy$

Intercept (Translation): $\frac{dTC}{db} = \int_{y \in Y_{BR}} \left(\frac{\partial TC}{\partial A^x(y)} - \frac{\partial TC}{\partial A^y(x)}\right) \frac{\partial A^x(y)}{\partial b} dy$

where $\frac{\partial A^x(y)}{\partial a} = -\frac{A^x(y)}{a}$ and $\frac{\partial A^x(y)}{\partial b} = -\frac{1}{a}$

· Gradient $\nabla TC = \left(\frac{dTC}{da}, \frac{dTC}{db}\right)^t$

· Hessian $\nabla^2 TC = \begin{bmatrix} \frac{\partial^2 TC}{\partial a^2} & \frac{\partial^2 TC}{\partial a \partial b} \\ \frac{\partial^2 TC}{\partial a \partial b} & \frac{\partial^2 TC}{\partial b^2} \end{bmatrix}$ from Proposition 4.9.

· **Check if $\nabla^2 TC$ is positive-definite. If yes $\nabla^2 TC := (\nabla^2 TC)^+$, else find smallest k where $(\nabla^2 TC)^+ := \nabla^2 TC + 4^k I$ is positive-definite.**

· Calculate direction vector for the slope(a) and the intercept(b).

$\begin{pmatrix} d_a^j \\ d_b^j \end{pmatrix} = -\left(\frac{\nabla TC}{(\nabla^2 TC)^+}\right)$ and normalize direction vector.

$$d_{SLOPE}^j = \frac{d_a^j}{\sqrt{(d_a^j)^2 + (d_b^j)^2}}$$

$d^j_{INTERCEPT} = \frac{d^j_b}{\sqrt{\left(d^j_a\right)^2 + \left(d^j_b\right)^2}}$

Step 3.2. Perform a line search for step size α^j

· Update the two allocation decisions $A1^j$ and $A2^j$

$(A1^j)' = \frac{A3 - \left(b^j - \alpha^j d^j_{INTERCEPT}\right)}{\left(a^j - \alpha^j d^j_{SLOPE}\right)}$

$(A2^j)' = \frac{A4 - \left(b^j - \alpha^j d^j_{INTERCEPT}\right)}{\left(a^j - \alpha^j d^j_{SLOPE}\right)}$

· Repeat **Steps 2.1, 2.2, 2.3** using $(A1^j)'$ and $(A2^j)'$

· Find $(\alpha^j)^* = \arg\min_{\alpha^j} TC$

Step 3.3. Update the allocation decisions

$A1^{j+1} = \frac{A3 - \left(b^j - \left(\alpha^j\right)^* d^j_{INTERCEPT}\right)}{\left(a^j - (\alpha^j)^* d^j_{SLOPE}\right)}$

$A2^{j+1} = \frac{A4 - \left(b^j - \left(\alpha^j\right)^* d^j_{INTERCEPT}\right)}{\left(a^j - (\alpha^j)^* d^j_{SLOPE}\right)}$

· Repeat **Steps 2.2, 2.3** using $A1^{j+1}$ and $A2^{j+1}$

· Return to **Step 3**

Step 4. Terminate with the solution $A1^j$, $A2^j$ and BR

Now, we provide an illustration of the above algorithm using the Squared Euclidean-metric based distance measure.

Example 4.6: Modified-Newton Improvement Algorithm - Squared Euclidean-metric (L_2^2) **Case**

For convenience, we use the same example in the preceding sections, where we have a square market region $\mathcal{M}$=$\{(x, y)|x \in (0, 100)$ and $y \in (0, 100)\}$,i.e. $M = 100$. We wish to determine an optimal allocation decision for a linear demand density function $(D(x, y) = 100 + 10x + 5y)$ over the market region $\mathcal{M}$. The starting solution for this instance is $A1$=35 and $A2 = 40$ at $y_{P1} = A3 = 40$ and $y_{P2} = A4 = 50$, respectively. The steps of the algorithm is as follows:

ITERATION 1

Step 1. Define and Initialize the model parameters and variables

$j = 0$

$\epsilon_{COST} := 1 \times 10^{-5}$

$y_{P1} = A3 = 40$ and $y_{P2} = A4 = 50$

$A1^{j=1}$=35 and $A2^{j=1} = 40$

$M = 100$

Step 2. Initialization: Allocate the service regions and optimally locate facilities

Step 2.1. Calculate Slope(a) and Intercept(b) of BR using $\left(\frac{y-A3}{A3-A4}\right) = \left(\frac{x-A1^j}{A1^j-A2^j}\right)$

Slope $a := 2$, Intercept $b := -30$

Step 2.2. Identify the following sets and functions given BR

$br(x) = y = 2x + 30$, $br(y)^{-1} = x = \frac{y}{2} + 15$

$Y_{BR} = [0, 100]$, $X_{BR} = [15, 65]$, $X_{A1} = [0, 15]$, $X_{A2} = [65, 100]$

Step 2.3. Find the optimal locations and calculate the total cost

Centroidal locations are optimal:

$(x_1^*, y_1^*)^{j=1} := (x_1^G, y_1^G) = (27.72, 69.07)$

$(x_2^*, y_2^*)^{j=1} := (x_2^G, y_2^G) = (73.27, 48.96)$

$TC^{j=1} = 8,757,663,304.33$

Step 3. Improvement: Update the first service region sizes $A1^{j=1}$ and $A2^{j=1}$

Do While $\left(\frac{|TC^{j+1} - TC^j|}{|TC^j|} \geq \epsilon_{COST}\right)$:

$j = j + 1$

Step 3.1. Calculate the partial gradients

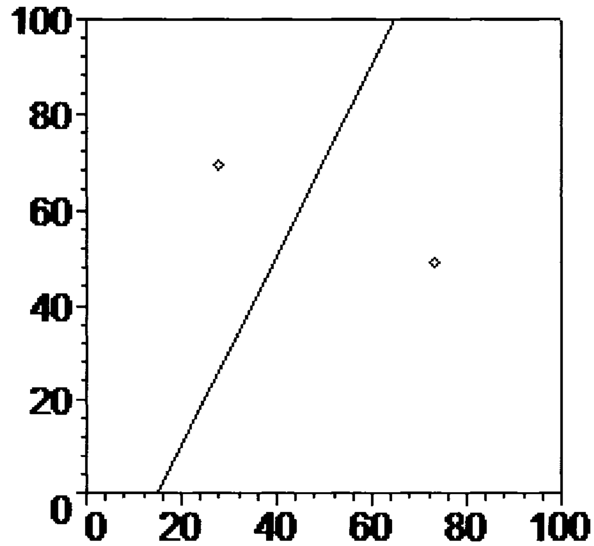

Figure 4.28: Starting solution of the Squared Euclidean-metric based example for Modified-Newton's improvement method (Example 4.6).

· Total cost with respect to single-dimensional allocation decisions $(A^x(y), A^y(x))$

Slope (Pure rotation): $\frac{dTC}{da}$ = 841,815,739.92

Intercept (Translation): $\frac{dTC}{db}$ = 20,032,429.78

$\nabla TC = (841,815,739.92, 20,032,429.78)^t$

· Hessian $\nabla^2 TC = \begin{bmatrix} 744{,}158{,}858.85 & 23{,}270{,}194.78 \\ 23{,}270{,}194.78 & 702{,}559.25 \end{bmatrix}$ is **not Positive-definite.**

· With **k:=9,**

$$(\nabla^2 TC)^+ := \nabla^2 TC + 4^{k=9}\mathbf{I} = \begin{bmatrix} 744{,}421{,}002.85 & 23{,}270{,}194.78 \\ 23{,}270{,}194.78 & 964{,}703.25 \end{bmatrix}$$

is Positive-definite

· Calculate direction vector for the slope(a) and the intercept (b).

$$\begin{pmatrix} d_a^{j=1} \\ d_b^{j=1} \end{pmatrix} = -\left(\frac{\nabla TC}{\left(\nabla^2 TC\right)^+}\right) = \begin{pmatrix} 1.9584 \\ -26.4751 \end{pmatrix}$$

Normalize the direction vector.

$d_{SLOPE}^{j=1} = 0.73771$

$d_{INTERCEPT}^{j=1} = -0.99728$

Step 3.2. Perform a line search for step size α^j

· Find $(\alpha^{j=1})^* = \arg\min_{\alpha^{j=1}} TC = 8.48$

Step 3.3. Update the allocation decisions

$A1^{j=2} = 44.78,\ A2^{j=2} = 52.05,\ TC^{j=2} = 8,532,834,684.62$

Table 4.6. illustrates the remainder of iterations. Optimality condition (0.001% improvement) is reached at $j = 5$ iteration. Iteration result for $j = 5$ is as below and displayed in Figure 4.29.

$$\begin{pmatrix} A1 \\ A2 \end{pmatrix}^{j=100} = \begin{pmatrix} 51.57 \\ 54.18 \end{pmatrix}$$

$$\begin{aligned} \mathbf{x}_1^* &= (x_1^*, y_1^*) = (32.66, 61.99) \\ \mathbf{x}_2^* &= (x_2^*, y_2^*) = (78.82, 49.93) \end{aligned}$$

$$\begin{aligned} br(x) &= y = 3.826x - 157.326 \\ TC &: = 8,459,944,242.44 \end{aligned}$$

		TRIGGERS				LOCATIONS				ALLOCATION LINE			
j	BR Type	A1	A2	A3	A4	x1	y1	x2	y2	Slope	Intercept	*TC*	*% Change*
0	I	35.00	40.00	40	50	27.72	69.07	73.27	48.96	2.000	-30.000	8,757,663,304.33	
1	I	44.78	52.05	40	50	38.08	70.13	76.56	43.16	1.374	-21.543	8,532,834,684.62	2.63%
2	I	45.20	49.59	40	50	32.07	65.99	76.79	48.11	2.277	-62.918	8,501,426,366.85	0.37%
3	I	50.84	54.33	40	50	33.68	63.62	78.77	48.57	2.858	-105.288	8,465,602,279.68	0.42%
4	I	51.43	54.07	40	50	32.62	62.05	78.77	49.91	3.793	-155.090	8,459,968,472.85	0.07%
5	I	51.57	54.18	40	50	32.66	61.99	78.82	49.93	3.826	-157.326	8,459,944,242.44	0.0003%

Table 4.6. Modified-Newton improvement algorithm's iteration results for the Squared Euclidean-metric (L_2^2) example (Example 4.6).

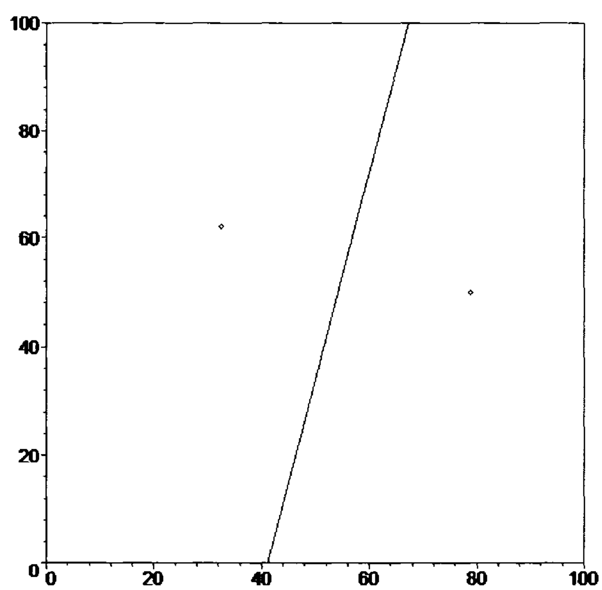

Figure 4.29: Solution at the end of 5^{th} iteration of the Squared Euclidean-metric based example for Modified-Newton's improvement method (Example 4.6).

In order to compare with the performance of first-order approach, steepest-descent, Table 4.7. illustrates iteration results for the same problem with L_2^2 and identical starting solution. When Table 4.6. and 4.7. are compared, it is obvious that second-order method is significantly better. This is due to the ill-conditioning of the Hessian of TC of the LAM-AVS. Steepest-descent, in Table 4.7., converges in more than 100 iterations, which is roughly twenty-fold worse than Newton's method. Since Newton's method require the calculation of Hessian, it is computationally costly. We have experimented with alternatives, such as Quasi-Newton and conjugate-gradient methods which perform significantly better than the first order approach. Conjugate-gradient method will be discussed in more detail in the next Chapter 5.

	TRIGGERS		ALLOCATION LINE		LOCATIONS								
j	A1	A2	Slope	Intercept	x1*	y1*	x2*	y2*	TC(j)	d_{SLOPE}	$d_{INTERCEPT}$	α	ΔTC(%)
1	35.000	40.000	2.000	-30.000	27.72	69.07	73.27	48.96	8,757,663,302.00	0.9997	0.0238	0.458	-
2	45.395	51.879	1.542	-30.011	36.59	69.01	76.92	44.50	8,521,668,250.00	-0.0238	0.9997	17.697	0.02695
3	44.676	49.770	1.963	-47.703	33.19	67.18	76.70	47.10	8,508,818,563.00	0.9997	0.0238	0.093	0.00151
4	46.892	52.238	1.870	-47.705	34.95	67.15	77.50	46.18	8,498,144,540.00	-0.0238	0.9997	12.660	0.00125
5	46.222	50.828	2.171	-60.362	33.07	66.15	77.20	47.58	8,491,555,991.00	0.9997	0.0238	0.073	0.00078
6	47.839	52.605	2.098	-60.364	34.30	66.12	77.81	46.98	8,485,744,834.00	-0.0238	0.9997	10.245	0.00068
7	47.239	51.510	2.341	-70.606	33.00	65.44	77.53	47.93	8,481,796,774.00	0.9997	0.0238	0.061	0.00047
8	48.493	52.877	2.281	-70.607	33.93	65.42	78.02	47.50	8,478,269,300.00	-0.0238	0.9997	8.570	0.00042
9	47.964	51.989	2.485	-79.175	32.95	64.92	77.76	48.20	8,475,731,300.00	0.9997	0.0238	0.051	0.00030
10	48.972	53.081	2.434	-79.176	33.69	64.90	78.16	47.87	8,473,446,321.00	-0.0238	0.9997	7.216	0.00027
11	48.507	52.345	2.606	-86.390	32.92	64.53	77.93	48.41	8,471,751,476.00	0.9997	0.0238	0.044	0.00020
12	49.333	53.236	2.562	-86.391	33.52	64.51	78.26	48.15	8,470,213,650.00	-0.0238	0.9997	6.387	0.00018
13	48.922	52.606	2.714	-92.776	32.89	64.21	78.06	48.59	8,468,991,391.00	0.9997	0.0238	0.038	0.00014
14	49.624	53.361	2.676	-92.777	33.39	64.19	78.34	48.38	8,467,880,100.00	-0.0238	0.9997	5.618	0.00013
15	49.260	52.819	2.809	-98.393	32.86	63.95	78.16	48.74	8,466,989,479.00	0.9997	0.0238	0.034	0.00011
16	49.860	53.463	2.776	-98.394	33.29	63.93	78.41	48.57	8,466,177,658.00	-0.0238	0.9997	-	0.00010
100	51.479	54.135	3.765	-153.842	32.68	62.08	78.79	49.88	8,459,958,200.00			-	

Table 4.7. Iteration results for Steepest-descent improvement algorithm for the case L_2^2 (Example 4.6).

4.5 Conclusions

In the context of the most general model, we first established the relation between solving the problem in the location variable space versus solving in the allocation variable space. The only approach that has been pursued in the literature is the former approach. In this chapter, we contribute by developing constructive and improvement based heuristics for the problem in the allocation variable space.

Our constructive heuristic approach builds on its counterpart in the single-dimensional case. In this approach, based on an initial shooting, we again tile the market region according to the first-order conditions. The main difference from its single dimensional counterpart is the boundary of the service region. On a line, the boundary is a point, on the other hand, in the planar setting, boundary is a straight line. Accordingly, when we shoot the first service region, we use the information from the first order conditions to match the second region with the market boundary. The level of match between the second service region and market boundary guides us in improving our shooting decisions.

We propose two improvement heuristics: steepest-descent (first-order method) and modified Newton's method (second-order method). In both of these heuristics, we improve the allocation decision, which is a straight line, by translation and rotation movements. Starting with an initial allocation solution, these methods improve the allocation decisions based on their cost improvement prospects of these two methods. One difference of these heuristics from the single dimensional improvement methods is that we are now iterating an allocation line rather than an allocation point.

In conclusion, with the constructive heuristic we try to find a solution which satisfies the first-order conditions whereas with the improvement heuristics we try to find a solution which will decrease the total cost. In addition, constructive approach can be viewed as an effort where we impose the equidistance condition and try to match with the market boundary. In contrast, improvement method imposes the market boundary and tries to achieve equidistance property of the allocation line with respect to locations. While the constructive heuristic constitutes an extension to the single-dimensional version, it is not straightforward to extend it further to the planar n-facility case as described in the next chapter. Reasons of this challenge will be discussed in more detail in the introduction section of the next chapter. On the other hand, we use most of the results developed in this chapter for the improvement heuristics in extension to the n-facility case.

Even though we have not included it in this chapter, for brevity, there is also another approach, the sequential location-allocation method which is originally proposed by (Cooper, 1964). This method so far has been used for the discrete demand cases. In the Appendix 4, we provide an extension of SLA for the continuous demand together with an illustration for a planar 2-facility example.

Chapter 5

Planar Model: N-Facility and Euclidean-Metric Case

5.1 Introduction

The purpose of this chapter is to develop solution methodologies for the planar n-facility location-allocation problems in the allocation variable space. This effort will extend the approaches developed for the 2-facility case. An example illustration of the 15-facility problem with nonlinear demand is provided in Figure 5.1. One major difference between the n-facility case and 2-facility case is the difficulty in the representation of the allocation decisions. As a result of this fundamental difference in the allocation variable representation, the adoption of the methods developed for the 2-facility case is not straightforward. Nonetheless, basic ideas underlying the methods developed for 2-facility case, such as the shape preserving transformation, help develop alternative methodologies for the n-facility case.

In this introduction section, Section 5.1., we discuss the challenge of moving from 2-facility representation of the allocation decisions to n-facility representation and the resulting difficulties in the adoption of the methods developed for the 2-facility case. We also describe briefly the voronoi-diagram approach as a tool for representing the allocation solutions. In Section 5.2., we introduce the additional notation such as the new decision variables: Vertices and edges. In Section 5.3., we extend the location-allocation problem formulations in location and allocation variable spaces (LAM-LVS and LAM–AVS) to the n-facility setting. Eventhough, there is no fundamental change neither in these models nor in their analytical properties, we still include them for the completeness of the material in this chapter. In Section 5.4., we provide a novel solution approach for the planar n-facility case with Euclidean-metric based distance measures. This approach is the vertex-iteration based improvement approach which extends the allocation decision improvement methods developed in the previous chapter. The only limitation of this approach is that it is suitable for mainly the Euclidean-metric based distance measures where the allocation decisions are in the form of straight lines. We first provide the vertex-based representation of the allocation decisions and then describe special procedures to handle the vertex-events which can cause inconsistencies. For the vertex-iteration based improvement approaches, we provide steepest-descent and conjugate-gradient solution procedures. In addition to formal representation of the algorithms, we provide example implementations for these two solution approaches.

In what follows we first briefly review voronoi diagram approach to the location-allocation problems. Next we discuss the challenges of extension of the solution techniques for 2-facility to n-facility case. More specifically, we will explain why the constructive solution technique cannot be adopted to the n-facility case and the necessary changes to the improvement based algorithms.

5.1.1 Voronoi Diagrams for Location-Allocation Problems

The purpose of this section is to summarize a powerful computational geometry approach to the location-allocation problem, namely the voronoi diagrams. This section is by no means a comprehensive overview of the voronoi applications; however, Okabe et al. (2000) represent a stellar reference on this subject. A whole chapter of this reference is dedicated to application of Voronoi dia-

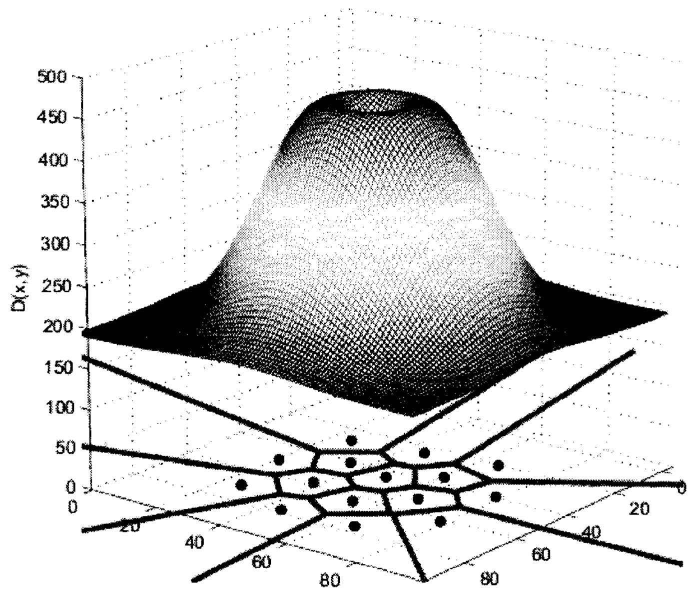

Figure 5.1: An illustration of the solution to the planar location-allocation problem with $n = 15$ service regions and a nonliner demand density function.

grams to various types of planar location problems. Voronoi diagrams are polygonal tesselations of the plane where each polygon ($\mathcal{A}_i$) is associated with a generator point (p_i). This association between the voronoi polygons and the generator points are based on a closeness measure such as a distance measure based on any metric. From this definition, a generator point (p_i) in a polygon ($\mathcal{A}_i$) is called the nearest generator. For notation reference, let's define the set of polygons as $\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2, ..., \mathcal{A}_n\}$ and the set of generator points as $P = \{p_1, p_2, ..., p_n\}$. An example voronoi diagram tesselation is illustrated in Figure 5.2.

Depending on the metric chosen as the basis of closeness measure, a voronoi polygon can be defined as collection of points satisfying the following relation which is based on the nearest-neighbor property. Note that this representation defines $\mathcal{A}_i$ as a closed set which is a subset of the plane where p_i dominates

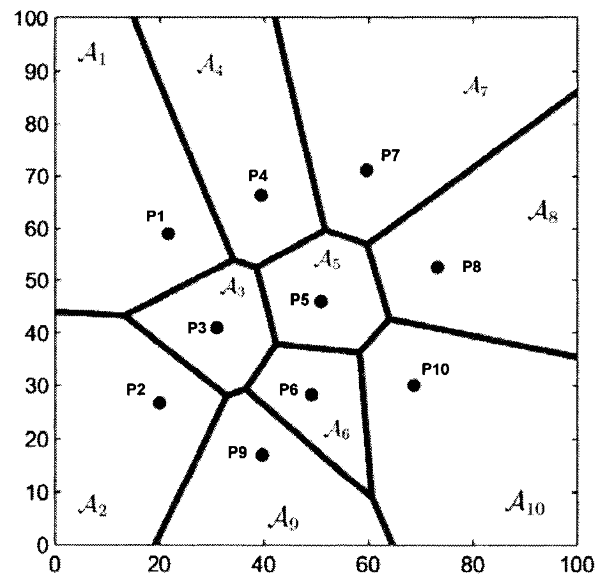

Figure 5.2: An illustration of voronoi-diagram for $n = 10$ generator points.

all other generators with respect to distance measure based on L_p norm.

$$\mathcal{A}_i = \bigcap_{i \neq j\ ,\ j=1,2\ldots n} \left\{ x \in R^2 | \quad ||p_i - x||_p \leq ||p_j - x||_p \right\}$$

It is also useful to define the set of **boundary points** of the $\mathcal{A}_i$ as $\partial\mathcal{A}_i$, which, then, allows us to define the intersection of boundaries of two voronoi polygons as follows.[8]

$$e_{ij} = \{x \in R^2 \mid x \in (\partial\mathcal{A}_i \cap \partial\mathcal{A}_j)\}$$

These intersection sets are generally referred as **Voronoi Edges** (e_{ij}). When three of these edges intersect at a point, V, then it is referred as **Voronoi Vertex**. When four or more edges intersect at a voronoi vertex then it is

[8]In planar 2-facility case, recall that, in place of $\partial\mathcal{A}_1 \cap \partial\mathcal{A}_2$, we used $\mathcal{A}_1 \cap \mathcal{A}_2$. Accordingly, allocation line BR of 2-facility case is a voronoi edge.

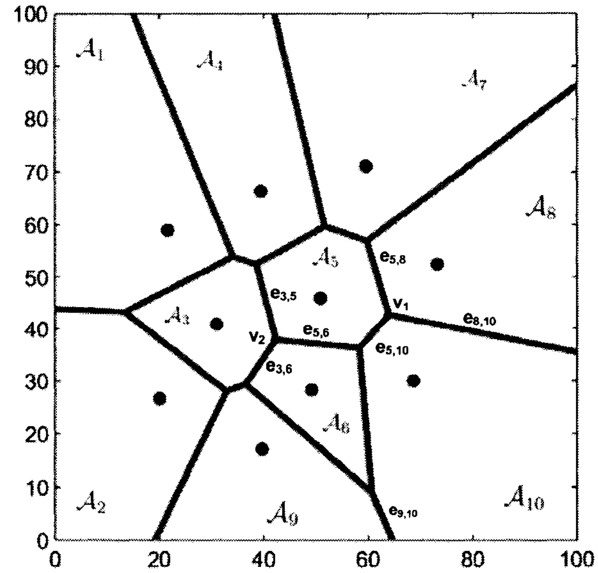

Figure 5.3: An illustration of the voronoi edges and vertices.

called a *degenerate* voronoi vertex. Figure 5.3 illustrates these definitions.

$$v = e_{ij} \cap e_{ik} \cap e_{jk}$$

There are interesting properties of Voronoi edges and vertices. From this point onward, we will be referring to voronoi diagrams based on Euclidean-metric based measures (i.e. L_2 and L_2^2). In the case of Euclidean-metric based distance measures, the voronoi edge, e_{ij}, is a bisector between p_i and p_j. Since each bisector divides the plane into two half spaces, a voronoi polygon as the intersection of at most $(n-1)$ half planes, where n is the number of generator points. Accordingly, a voronoi polygon is bounded by at most $(n-1)$ voronoi edges and $(n-1)$ voronoi vertices. Note that voronoi polygons can be unbounded, such as A_1, A_1 and A_7 in Figure 5.3. Unless the generator points are collinear, the voronoi edges are either line segments or half lines. In addition, Voronoi diagram of n generators has at most $(2n-5)$ vertices and

$(3n-6)$ edges.

5.1.2 Extension from 2-facility to n-facility

This section serves two purposes: Discussion of the difficulty of developing a constructive solution approach and justification of the need for the modifications to the improvement algorithms introduced for the 2-facility case in the previous chapter.

In both the single-dimensional case and the 2-facility planar cases, the constructive solution approach follows the following generic steps: Generate initial trigger(s), solve sequentially the differential equations in order to tile the remaining service regions, and improve the initial triggers according to the violation of boundary conditions. The challenge in implementing this constructive solution approach to the n-facility planar case is that the number of remaining service regions at each trigger level is unknown. Please see the $y = y_1$ and $y = y_2$ in Figure 5.4, which illustrates this point. Note that this is a mute issue in the planar 2-facility case.

We now turn to the improvement based heuristic. In the planar 2-facility case, we have described that there are two ways to improve an allocation decision: Pure rotation and translation. Recall that the allocation decision is a straight line for the Euclidean-metric cases and a special form composed of at most three segments for the Manhattan-metric case, which will be discussed in Chapter 7. Furthermore, we showed that with these shape preserving transformations, it is possible iterate from one feasible solution to the other by changing the slope and intercept of the allocation line. In the 2-facility case, this is a viable procedure since we have a single allocation line BR which characterizes the solution uniquely. However this is a challenging task when there

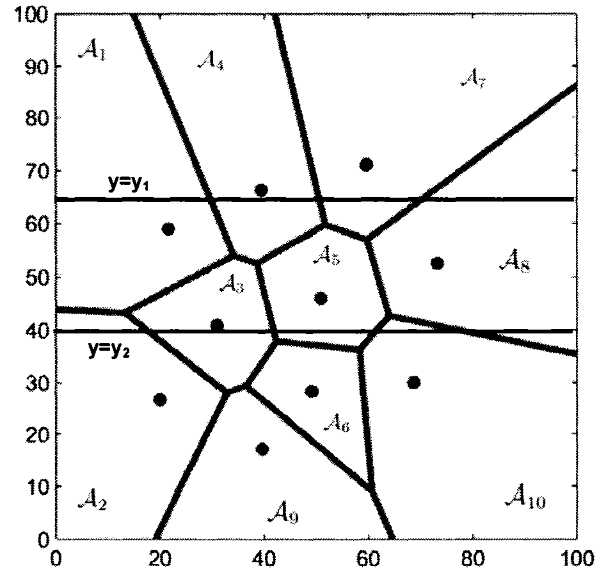

Figure 5.4: In the n-facility case, the major limitation for constructive solution approaches is the inability to know the number of service regions at each trigger level.

are three or more service regions. Without loss of generality, we will illustrate the arising complication for the Euclidean-metric based case. Figure 5.5 helps us understand this further.

In Figure 5.5, the solid lines depict a feasible allocation solution for 3-facility case. In the terminology of the previous chapter, we have three allocation lines, $BR1$, $BR2$, and $BR3$, which allocate the rectangular market region to three service regions. The vertex point, P_{123}, is the point at which three allocation decisions intersect. Improvement algorithms presented in the 2-facility chapter is based on the shape-preserving transformation of the allocation line. Accordingly, when we improve the allocation lines, $BR1$, $BR2$, and $BR3$, independently to reduce total cost, as shown in Figure 5.5 with dashed lines, we would end up with three candidates points (P_{12}, P_{23}, and P_{13}) to replace P_{123}. We need to evaluate these three alternative cases and choose

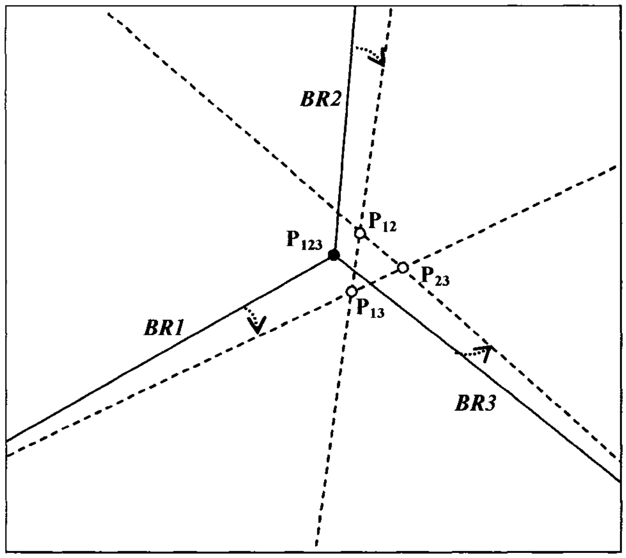

Figure 5.5: Iterating allocation decisions ($BR1, BR2$ and $BR3$).

the best as the next solution. None of these candidate points (P_{12}, P_{23}, and P_{13}) conform to the definition of a vertex. In order for P_{12}, P_{23}, or P_{13} to be considered as a vertex, we need to force the third allocation line to pass through that point.[9]

There are other alternative ways if one wants to iterate the allocation decisions independently. One approach is to choose P_{123} as the axis of rotation (i.e. reference point of rotation which is kept constant) and rotate allocation decisions around that point. Another approach is to rotate only two allocation decisions (i.e. $BR1$ and $BR2$) and consider the effect of their rotation on the remaining one ($BR3$). Whereas the first method would be suboptimal since we are restricting the search space (i.e. any point in the region could be a

[9]An independent iteration of each edge would create 3 candidate vertices. Hence, we have to search among, at most, $\binom{3\,(2n-5)}{2n-5}$ possible feasible solutions to evaluate an improvement iteration step. As an example when we have $n = 10$ facilities, the voronoi tesselation will have at most 15 ($= 2 \times 10 - 5$) vertices. On the other hand, when we iterated the allocation lines (i.e. voronoi edges), there will be $\binom{3\,(15)}{15} = 1.4771 \times 10^{13}$ possible solutions to evaluate and choose from.

vertex point), second method would be as difficult to implement as independent iteration of the allocation decisions. For instance the choice of which two allocation decisions to improve and which to leave as the dependent allocation decision constitutes another set of combinatorial decisions.

In the light of these complications, we decided to move the vertices rather than the allocation lines. We call this **vertex-iteration based improvement approach** for the n-facility case. This not only helps us overcome the above difficulties but also allows us to use most of the results obtained in the 2-facility case.

5.2 Description of Parameters and Notation

In what follows, we first describe parameters and decision variables of the planar location-allocation problems with more than two facilities. For the continuous flow of the material in this chapter, some of the previously defined notation will be repeated here. Some of these definitions could be observed in Figure 5.3, where an $n = 10$ facility solution is presented.

This section uses the following notation and parameter definitions:

Parameters:

$\mathbf{x}$: a point in the two dimensional space $\mathbf{x} \equiv (x, y)$

$\mathcal{M}$: Two dimensional market area (assumed to be a closed and compact set)

$D(\mathbf{x})$: Demand density function over the two-dimensional market region $\mathcal{M}$ $(D(\mathbf{x}){\equiv}D(x, y))$

$d_p(\mathbf{x}_i, \mathbf{x})$: Shortest distance between $\mathbf{x}_i$ and $\mathbf{x}$ for a given distance measure p

($\mathbf{p} = \mathbf{L}_2$ denotes *Euclidian-metric*, $\mathbf{p} = \mathbf{L}_2^2$ denotes *Squared Euclidian-metric*)

As in the 2 facility case, we have two main decision variables: *Location* decisions and *Allocation* decisions. These decision variables are defined below.

Decision Variables:

$\mathbf{x}_i$: locational coordinates of the facility in service region i, i.e. $\mathbf{x}_i \equiv (x_i, y_i)$.

$\mathcal{A}_i$: Allocation polygon for i^{th} service region (assumed to be a closed set and in polygonal shape but not necessarily convex).

$\mathbf{x}_i^*$: Optimal locations given the allocation decisions $\mathcal{A}_{i=1,2,..,n}$

In the n-facility case, we need to introduce additional constructs for allocation decisions which are the **vertices** and **edges** forming the allocation decisions or more precisely allocation polygons.

e_{ij} : is the edge between two service regions $\mathcal{A}_i$ and $\mathcal{A}_j$ and more formally $e_{ij} = \mathcal{A}_i \cap \mathcal{A}_j$

v_k : is the vertex formed by the intersection of at least three edges (i.e. $v_k = e_{ij} \cap e_{ik} \cap e_{jk}$)

In the 2-facility case, a single allocation line BR is sufficient to characterize the allocation decisions $\mathcal{A}_1$ and $\mathcal{A}_2$. Each of these allocation decisions are bounded with the allocation line and market boundary. In comparison with the 2-facility case's allocation line BR, in the n-facility case we have a finite number of edges, each of which is an allocation line separating the two service regions sharing that edge(s). Note that it is possible to have more than one edge shared by the same pair of allocation polygons if they are non-convex polygons. Therefore, in order to characterize an allocation polygon, we need to characterize all the edges which separate the polygon from its neighbors.

Since within each allocation polygon $\mathcal{A}_i$ we can draw horizontal and vertical straight lines from one edge to another edge, we can define the **Single-dimensional Allocation Decisions** ($A_i^y(x)$ and $A_i^x(y)$) as in the 2 facility case. The subscript i denotes that these allocation decisions are measured within the allocation polygon $\mathcal{A}_i$. For notational accuracy, we will define these single dimensional decisions more formally. For these definitions Figure 5.6 is illustrative. In the 2-facility case, since there is a single allocation line BR and that single dimensional allocation decisions were sufficient to characterize this allocation line, these single-dimensional allocation decisions are measured from two reference axes, i.e. $x = 0$ and $y = 0$. In n-facility case, however, there allocation decisions are formed by more than one edge (a.k.a allocation line), thus single dimensional allocation decisions for an allocation polygon $\mathcal{A}_i$ need to be measured from variable reference points as illustrated in Figure 5.6. Let's take the x-dimensional allocation decision at level y, namely $A_i^x(y)$. This single-dimensional allocation decision would be measured from $B_i^x(y)$, the starting boundary point of $\mathcal{A}_i$ at the level y. Similarly $B_i^y(x)$ denotes the starting boundary point of $\mathcal{A}_i$ at the level x for the single-dimensional allocation decision $A_i^y(x)$. If we denote the set of indices of the service regions $\mathcal{A}_{i=1,2,\ldots n}$ at x and y levels as $OS(x)$ and $OS(y)$, then we could define the starting boundary points with respect to these sets.

$OS(x), OS(y)$: set of indices of the service regions $\mathcal{A}_{i=1,2,\ldots n}$ at x and y levels.

Note that the set of indices are ordered according to the corresponding service regions position with respect $x = 0$ and $y = 0$. For example, if a straight line parallel to $x - axis$ at level $y = y_1$ passes through both $\mathcal{A}_i$ and $\mathcal{A}_j$ and if $\mathcal{A}_i$ is closer to $x = 0$ than $\mathcal{A}_j$ then in the set $OS(y = y_1)$, i will precede j. In order to refer to this ordering, we define $pos(i, S) < pos(j, S)$, where $pos(\cdot, S)$ is defined as below.

$pos(k, S)$: represents the position of element k in set S.

Now we could define the starting boundary points, $B_i^x(y)$ and $B_i^y(x)$, for the service region $\mathcal{A}_i$ at any level of x and y-dimensions.

$$B_i^x(y) = \sum_{\substack{j \in OS(y) \\ pos(j,OS(y)) < pos(i,OS(y))}} A_j^x(y)$$

$$B_i^y(x) = \sum_{\substack{j \in OS(x) \\ pos(j,OS(x)) < pos(i,OS(x))}} A_j^y(x)$$

For notational brevity, we herein limit the use of the $B_i^x(y)$ and $B_i^y(x)$ and assume that single dimensional allocation decisions $A_i^x(y)$ and $A_i^y(x)$ would be interpreted according to the starting boundary points within each service region.

Observe that, in the above notation, we take $x = 0$ and $y = 0$ as the reference axes and the increasing direction from the starting boundary points, $B_i^x(y)$ and $B_i^y(x)$ as the measure of the single dimensional allocation decisions $A_i^x(y)$ and $A_i^y(x)$. However since the results in this chapter can be shown to be adopted for any reference axes and directions, there is no loss of generality of our results with respect to the above definitions and assumptions.

Lastly, we define the following objective functions, overall problem and for service region $\mathcal{A}_i$.

$TC(A_i^y(x), A_i^x(y), \mathbf{x}_i, i = 1, 2, ..., n)$: total cost objective function for a

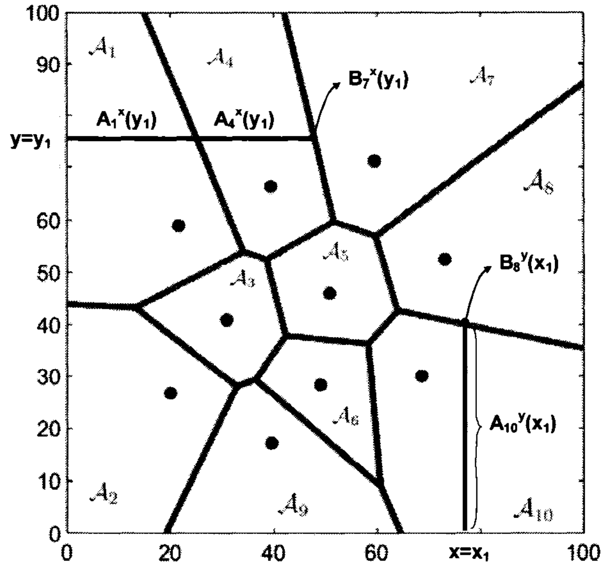

Figure 5.6: Illustration of the single-dimensional allocation decisions and the boundary points.

given solution of $(A_i^y(x), A_i^x(y), \mathbf{x}_i, i = 1, 2, ..., n)$

$TC_i(A_i^y(x), A_i^x(y), \mathbf{x}_i)$: cost in the service region $\mathcal{A}_i$.

Accordingly following holds true.

$$TC(A_i^y(x), A_i^x(y), \mathbf{x}_i, i = 1, 2, ..., n) = \sum_{i=1,2,...n} TC_i(A_i^y(x), A_i^x(y), \mathbf{x}_i)$$

5.3 Alternative Modeling Approaches for N-Facility Case

In this section, we will extend the two alternative modeling approaches for the 2-facility planar location-allocation problems of Chapter 4 to n-facility case. We will again introduce the location-allocation problem modeling in two different variable spaces, **Location Variable Space (LVS)** and **Allocation Variable Space (AVS)**. In all of these models, we use $(\mathbf{x}_{i=1,2,...,n})$ to denote the location decisions and $(A_i^y(x),\ A_i^x(y))$ to denote the single-dimensional allocation decisions *within* each allocation polygon. For sake of notational simplicity, we use implicit representation of the objective function in this section. In the following sections, individual terms of the objective function will be specified in more detail.

The generic model in the joint variable space is as follows.

Location-Allocation Model (LAM):

$$\min_{\substack{A_i^x(y), A_i^y(x) \\ \mathbf{x}_i=(x_i,y_i)}} TC(A_i^y(x), A_i^x(y), \mathbf{x}_i, i = 1, 2, ..., n)$$

subject to

$$(B_i^x(y) + A_i^x(y), y) = (x, B_i^y(x) + A_i^y(x)) \qquad for\ \ (x, y) \in \partial\mathcal{A}_i$$

Since the feasible region is composed of affine relations, LAM is a biconvex programming problem; when we fix the allocation decisions, it becomes a multifacility planar location problem and fixing location decisions transforms it into a transportation problem. It can be further shown that LAM is non-convex problem for L_2 and L_2^2.

Next section analyzes the model in the location variable space, where the allocation decisions are optimized given the location decisions.

5.3.1 Modeling in Location Variable Space

A location prioritized model is based on transforming LAM into an equivalent form with decisions variables as the location decisions *only*. At this point, we assume that our total cost is solely composed of the transportation cost.

Therefore the following problem is equivalent to the LAM:

$$\min_{\mathbf{x}_{i=1,2,...n}} \widetilde{TC}(\mathbf{x}_i, i = 1, 2, ..., n)$$

where

$$\begin{aligned}\widetilde{TC}(\mathbf{x}_i, i &= 1, 2, ..., n) \\ &= \min_{\substack{A_i^y(x), A_i^x(y) \\ i=1,2...,n}} \left\{ \begin{array}{c} TC(A_i^y(x), A_i^x(y), \mathbf{x}_i, i = 1, 2, ..., n) | \\ (B_i^x(y) + A_i^x(y), y) = (x, B_i^y(x) + A_i^y(x)) \\ \forall\, (x, y) \in \partial \mathcal{A}_i \end{array} \right\}\end{aligned}$$

The above bi-level optimization is equivalent to optimizing first over the allocation decisions and then over the location decisions. If the allocation decisions $A_i^y(x)$ and $A_i^x(y)$ were possible to express in closed form in terms of the location decisions then $\widetilde{TC}(\mathbf{x}_i, i = 1, 2, ..., n)$ could have been optimized directly. Since this is not possible, we include the solution to the allocation-problem as below in the constraint set of the location problem.

$$d_p\left(\mathbf{x}_i, (x, B_i^y(x) + A_i^y(x))\right) = d_p\left(\mathbf{x}_j, (B_i^x(y) + A_i^x(y), y)\right) \tag{66}$$

$$\begin{aligned}\text{for } \forall (B_i^x(y) + A_i^x(y), y) = (x, B_i^y(x) + A_i^y(x)) \;\; \in \partial \mathcal{A}_i \cap \partial \mathcal{A}_j \\ \text{for } \forall i,j, \text{ and } i \neq j\end{aligned} \tag{67}$$

We now present the *location-allocation problem in the location variable space.*

LAM- Location Variable Space (LAM-LVS):

$$\min_{\mathbf{x}_{i=1,2,...n}} TC(A_i^y(x), A_i^x(y), \mathbf{x}_i, i = 1, 2, ..., n)$$

$s.t.$

$$(B_i^x(y) + A_i^x(y), y) \quad = \quad (x, B_i^y(x) + A_i^y(x)) \;\; for \; \forall\,(x,y) \in \partial\mathcal{A}_i \text{ and } \forall i$$

$$\text{and constraint} \quad (66)$$

The only difference between LAM and LAM-LVS is the last constraint (66) which conditions the optimality of allocation decisions on the location decisions, while making the allocation decisions endogenous decision variables and leaving the location variables as exogenous decision variables. Since LAM-LVS and LAM are equivalent problems, LAM-LVS is also a nonconvex problem for the n-facility case. The first order necessary conditions for LAM-LVS in the case of L_2 and L_2^2 for n-facility case are analogous to the 2-facility case. For completeness we simply state their n-facility version with the notation used in this chapter.

Proposition 5.1.

The optimal locations of the n-facilities $(\mathbf{x}_i^*)$*, given the allocation decisions, satisfy the following conditions when the distance measure is based on the* **Squared Euclidean – Metric** $(\mathbf{L}_2^2)$:

$$(x_i^*, y_i^*) = \left(x_i^G, y_i^G\right) \qquad \text{for } i = 1, 2, ..., n$$

where x_i^G *and* y_i^G *are the* $x-$ *and* $y-$ *dimensional centroids of* $A_{i=1,2,...,n}$ *with respect to* $D(\mathbf{x})$.

$$x_i^G = \frac{\int\limits_{\mathcal{A}_i} xD(\mathbf{x})d\mathbf{x}}{\int\limits_{\mathcal{A}_i} D(\mathbf{x})d\mathbf{x}} \qquad \text{and} \qquad y_i^G = \frac{\int\limits_{\mathcal{A}_i} yD(\mathbf{x})d\mathbf{x}}{\int\limits_{\mathcal{A}_i} D(\mathbf{x})d\mathbf{x}} \qquad \text{for } i = 1, 2, ..., n$$

Proof.

Proof follows from the 2-facility case proposition 4.1. ■

Proposition 5.2.

The optimal locations of the n-facilities ($\mathbf{x}_i^*$)*, given the allocation decisions, satisfy the following conditions when the distance measure is based on the* **Euclidean − Metric** $(\mathbf{L}_2)$:

$$\begin{aligned}
\int_{\mathcal{A}_i} \frac{(x_i^* - x)}{||\mathbf{x}_i^* - \mathbf{x}||} D(\mathbf{x}) d\mathbf{x} &= 0 \qquad \text{for } i = 1, 2, ..., n \\
\int_{\mathcal{A}_i} \frac{(y_i^* - y)}{||\mathbf{x}_i^* - \mathbf{x}||} D(\mathbf{x}) d\mathbf{x} &= 0 \qquad \text{for } i = 1, 2, ..., n
\end{aligned}$$

Proof.

Proof follows from the 2-facility case proposition 4.2. ■

5.3.2 Modeling in Allocation Variable Space

Similar to the location-allocation model in the location variable space, the model in allocation variable space is also based on transforming LAM into an equivalent form with decisions variables as the allocation decisions *only*. Again for a given number of facilities, without loss of generality, we assume that our total cost is composed of only the transportation costs. Hence, the following problem is equivalent to the LAM.

$$\min_{A_i^y(x), A_i^x(y)} \widetilde{TC}(A_i^y(x), A_i^x(y))$$

where

$$\widetilde{TC}(A^x(y), A^y(x)) = \min_{\mathbf{x}_{i=1,2,\dots,n}} \{TC(A_i^y(x), A_i^x(y), \mathbf{x}_i, i = 1, 2, ..., n)\} \tag{68}$$

Solving the above bi-level model is equivalent to optimizing first over the location decisions and then over the allocation decisions. Again the optimal solution to the single facility location problem in (68) can be expressed in closed form for only the single-dimensional case. These optimal location solutions satisfy the first order necessary conditions of the LAM-LVS in the previous section and, thus, when we include them in the constraint set of the LAM, we obtain the following *location-allocation model in the allocation variable space.*

LAM- Allocation Variable Space (LAM-AVS):

$$\min_{\substack{A_i^y(x), A_i^x(y) \\ i=1,2,\dots,n}} TC(A_i^y(x), A_i^x(y), \mathbf{x}_i^*, i = 1, 2, ..., n)$$

$$s.t.$$

$$\begin{aligned} (B_i^x(y) + A_i^x(y), y) &= (x, B_i^y(x) + A_i^y(x)) \qquad for\ \forall\, (x, y) \in \partial\mathcal{A}_i \text{ and } \forall i \\ \mathbf{x}_i^* &= \arg\min_{(\mathbf{x}_i)} \int_{\mathcal{A}_i} d_p(\mathbf{x}_i, \mathbf{x}) D(\mathbf{x}) d\mathbf{x} \quad \text{for } i = 1, 2, ..., n \end{aligned} \tag{69}$$

Since LAM-AVS and LAM are equivalent problems, LAM-AVS is also a non-convex problem. First order necessary conditions for LAM-AVS in n-facility case are in the same form for both distance measures (i.e. L_2 and L_2^2).

First Order Necessary Conditions for LAM-AVS (for $p = L_2$, and L_2^2) :

$$d_p\left(\mathbf{x}_i, (x, B_i^y(x) + A_i^y(x))\right) - d_p\left(\mathbf{x}_j, (B_i^x(y) + A_i^x(y), y)\right) = 0 \tag{70}$$

for $\forall (B_i^x(y) + A_i^x(y), y) = (x, B_i^y(x) + A_i^y(x)) \in \partial\mathcal{A}_i \cap \partial\mathcal{A}_j$ and $\forall i,j,\ i \neq j$.

In the next section, we describe two improvement based solution methods to determine the local optimum solutions for the LAM-AVS. These are **Steepest-Descent Improvement** and **Conjugate-Gradient Improvement** methods which are based on the iteration of vertices.

5.4 Solution Methodologies

As stated before, a direct adaptation of the allocation line improvement methods developed for the 2-facility case is impossible due to the combinatorial aspect of choosing the best feasible vertex. Feasible vertex is defined as the intersection point of three or more allocation edges. In order to avoid the need to choose between alternative vertices, we implement a different approach where we iterate the vertices rather than the allocation lines (i.e. edges). Note that iteration of vertices is impossible without iteration of the allocation lines, but the difference is the need for choosing between candidate vertices, illustrated in Figure 5.5, is eliminated when vertices are used as the iteration variables. In what follows, we first present some analytical results for the vertex-representation. Next, we illustrate some special vertex events and corresponding handling methods. Lastly we present two improvement based solution procedures.

5.4.1 Vertex-based Representation

In this section, we first describe the representation of feasible allocation solutions using vertices of the allocation polygons and describe three types of vertices. Next, we derive first-order differential relationships between the vertices and edges of the allocation polygons. Using these results and the results from the planar 2-facility chapter, we derive gradients of the objective function with respect to the vertices. Finally, we briefly explain the special cases of the vertex events as a result of vertex iterations and how we handle these vertex events in our improvement algorithms.

In the traditional voronoi diagram approach, there are two types of vertices: a set of vertices formed by the intersection of voronoi edges and a single vertex which is located at infinity. In contrast, we here define three alternative types of vertices: **Interior** vertices, **Border** vertices, and **Market-boundary** vertices. Interior vertices are the ones formed by the intersection of three or more edges of the allocation polygons. Border vertices are formed by the intersection of market boundary and one or more edges of the allocation polygons. Market boundary vertices are the ones which are formed by the intersection of two edges of the market-boundary. Without loss of generality, we herein assume a square-shaped market region $\mathcal{M}$. Figure 5.7 illustrates these three different types of vertices for a 3-facility example.

In this example there is only one interior vertex, v_1, which is formed by the intersection of three edges (e_{12}, e_{13}, e_{23}) as shown with filled circle. Note that this is a non-degenerate vertex since it is formed by the intersection of exactly three edges. In addition, there are three border vertices, namely v_2, v_3, and v_4,

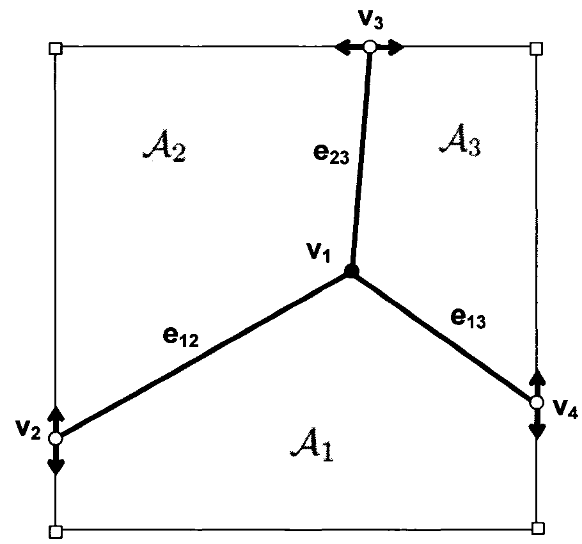

Figure 5.7: Illustration of three vertex types: Interior, Border and Market-boundary vertices.

as shown with empty circles. Finally there are four market-boundary vertices, namely the corners of the market boundary as shown with empty squares.

In this example, it is possible to cover the entire solution space for 3-facility case by iterating the four vertices, one interior and three border vertices, simultaneously. The main difference between an interior vertex and a border vertex is the admissible movement direction. Whereas an interior vertex is allowed to move on any directional vector, border vertices can only move on either on x-dimensional or y-dimensional vectors. These restrictions are illustrated in Figure 5.7 with arrows. For example, border vertices v_2 and v_4 can only translate along the y-coordinate axis, whereas v_3 can only translate horizontally. Market-boundary vertices are static and thus not allowed to iterate. Note that, even though it is visually possible for a vertex to appear as both a border and a market-boundary vertex, we treat it as a border vertex which is allowed to move on either x-axis or y-axis but not both.

Next, we will illustrate the differential relationships between the vertices and edges of the allocation polygons using the example in Figure 5.7. Let's define the following slope and intercept parameters for the three edges and the coordinates of the vertices:

a_{ij} : slope of the edge e_{ij} $(= \partial \mathcal{A}_i \cap \partial \mathcal{A}_j)$

b_{ij} : intercept of the edge e_{ij} $(= \partial \mathcal{A}_i \cap \partial \mathcal{A}_j)$

(v_k^x, v_k^y) : x- and y-coordinates of the vertex v_k.

Recall that from Proposition 4.4. of the previous chapter we have the following first order relations between single dimensional allocation decisions and the slope when the allocation line is rotated around **a reference axis** (x_r, y_r).

$$\frac{dA^x(y)}{da_r} = \frac{(y_r - y)}{a^2} \tag{71}$$

$$\frac{dA^y(x)}{da_r} = (x - x_r) \tag{72}$$

Based on these results, we now provide the following proposition which establishes the relationship between the movement of a vertex and the slope and intercept of one of its edges.

Proposition 5.3.

Suppose an edge $\mathbf{e}_{ij}$ *passes through two vertices* $\mathbf{v}_k = (v_k^x, v_k^y)$ *and* $\mathbf{v}_t = (v_t^x, v_t^y)$*. Moving the vertex* $\mathbf{v}_k$ *by increasing* v_k^x *and* v_k^y*, would change the slope* (a_{ij}) *and intercept* (b_{ij}) *of the edge* $\mathbf{e}_{ij}$ *according to the following relations:*

$$\frac{da_{ij}}{dv_k^x} = \frac{a_{ij}^2}{(v_t^y - v_k^y)} \qquad \textit{and} \qquad \frac{da_{ij}}{dv_k^y} = (v_k^x - v_t^x)$$

$$\frac{db_{ij}}{dv_k^x} = \frac{a_{ij}^2 v_t^x}{(v_k^y - v_t^y)} \qquad and \qquad \frac{db_{ij}}{dv_k^y} = \frac{(v_t^x - v_k^x)}{v_t^x}$$

Proof.

Proof is provided in Appendix 5. ■

Using above relations and Figure 5.7, we now exemplify the first-order relationships between two vertex types and the allocation lines (i.e. edges of the allocation polygon). First, we provide the results for the interior vertex, v_1. A differential change in the coordinates of v_1 would change the slope and intercept of the three edges (e_{12}, e_{13}, e_{23}). Let's focus on the edge e_{12}. Note that vertex v_2 is fixed. A differential change in the x-coordinate of vertex v_1 is equivalent to changing the single dimensional allocation decision $A^x(y = v_1^y)$ with respect to the reference axis (v_2^x, v_2^y):

$$\frac{\partial a_{12}}{\partial v_1^x} = \frac{a_{12}^2}{(v_2^y - v_1^y)}$$

Similarly, a differential change in the x-coordinate of vertex v_1 is equivalent to changing the single dimensional allocation decision $A^y(x = v_1^x)$ with respect to the reference axis (v_2^x, v_2^y):

$$\begin{aligned} \frac{\partial b_{12}}{\partial v_1^x} &= \frac{a_{12}^2 v_2^x}{(v_2^y - v_1^y)} \\ &= 0 \text{ since } v_2^x = 0 \end{aligned}$$

Similar analysis could be performed for the y -coordinate of v_1, i.e. v_1^y :

$$\begin{aligned}
\frac{\partial a_{12}}{\partial v_1^y} &= \frac{1}{(v_1^x - v_2^x)} = \frac{1}{v_1^x} \\
\frac{\partial b_{12}}{\partial v_1^y} &= \frac{-v_2^x}{(v_1^x - v_2^x)} = 0
\end{aligned}$$

Similar type of analysis on e_{13} would yield the following results:

$$\begin{aligned}
\frac{\partial a_{13}}{\partial v_1^x} &= \frac{a_{13}^2}{(v_4^y - v_1^y)} \qquad and \qquad \frac{\partial b_{13}}{\partial v_1^x} = \frac{a_{13}^2 M}{(v_4^y - v_1^y)} \\
\frac{\partial a_{13}}{\partial v_1^y} &= \frac{1}{(v_1^x - M)} \qquad and \qquad \frac{\partial b_{13}}{\partial v_1^y} = \frac{-M}{(v_1^x - M)}
\end{aligned}$$

Similar type of analysis on e_{23} would yield the following results:

$$\begin{aligned}
\frac{\partial a_{23}}{\partial v_1^x} &= \frac{a_{23}^2}{(M - v_1^y)} \qquad and \qquad \frac{\partial b_{23}}{\partial v_1^x} = \frac{a_{23}^2 v_3^x}{(M - v_1^y)} \\
\frac{\partial a_{23}}{\partial v_1^y} &= \frac{1}{(v_1^x - v_3^x)} \qquad and \qquad \frac{\partial b_{23}}{\partial v_1^y} = \frac{-v_3^x}{(v_1^x - v_3^x)}
\end{aligned}$$

Finally, we illustrate first-order relationship between a border vertex and its corresponding allocation polygon edge. In the example shown on Figure 5.7, all the three border vertices connects to only one edge of the allocation polygons. We here choose v_3 and perform similar first-order analysis as above. Note that v_3 is only allowed to translate horizontally (i.e. v_3^x is variable and $v_3^y = M$ is constant), and the reference axis is vertex v_1, (v_1^x, v_1^y).

$$\frac{\partial a_{23}}{\partial v_3^x} = \frac{a_{23}^2}{(v_1^y - M)} \qquad and \qquad \frac{\partial b_{23}}{\partial v_3^x} = \frac{a_{23}^2 v_1^x}{(v_1^y - M)}$$

In summary, for a given allocation solution with allocation polygons $\mathcal{A}_{i=1,2\ldots n}$, the first order relationship between the vertices (v_k) and their corresponding edges (e_{ij}) can be calculated through relations in Proposition 5.3. Note that for border vertices, either $\left(\frac{da_{ij}}{dv_k^x} \text{ and } \frac{db_{ij}}{dv_k^x}\right)$ or $\left(\frac{da_{ij}}{dv_k^y} \text{ and } \frac{db_{ij}}{dv_k^y}\right)$ is used.

We now present the first-order relationships between the vertices and the objective function $TC = TC(A_i^y(x), A_i^x(y), \mathbf{x}_i, i = 1, 2, ..., n)$:

$$\frac{\partial TC}{\partial v_k^x} = \sum_{\forall e_{ij} \in E_k} \left(\frac{\partial TC}{\partial a_{ij}} \frac{\partial a_{ij}}{\partial v_k^x} + \frac{\partial TC}{\partial b_{ij}} \frac{\partial b_{ij}}{\partial v_k^x} \right) \tag{73}$$

$$\frac{\partial TC}{\partial v_k^y} = \sum_{\forall e_{ij} \in E_k} \left(\frac{\partial TC}{\partial a_{ij}} \frac{\partial a_{ij}}{\partial v_k^y} + \frac{\partial TC}{\partial b_{ij}} \frac{\partial b_{ij}}{\partial v_k^y} \right) \tag{74}$$

where E_k is the set of edges that intersect at vertex v_k.

The partial derivatives of the TC with respect to the slope (a_{ij}) and intercept (b_{ij}), $\frac{\partial TC}{\partial a_{ij}}$ and $\frac{\partial TC}{\partial b_{ij}}$, have been derived in the previous chapter for the planar 2-facility case, and thus, not repeated here. Corresponding formulae can be found in Propositions 4.7. and 4.8. as (59), (60) and (61). The derivative information in (73) and (74) is calculated for each interior and border vertex and used in the two improvement algorithms described in the next section.

In the next section, we describe the special vertex events that arise as a result of changes in the vertex coordinates. For brevity we will simply demonstrate the most important five of them in this section.[10]

5.4.2 Special Vertex-events and Event-Handling

For an allocation solution $\mathcal{A}_{i=1,2,..,n}$ to be feasible the following two conditions must be met: Allocation decisions should cover the market region (75) and

[10] There are nine vertex events which we handle in our solution approach. The remaining four events are special cases when two, of the first five cases, occur at the same time.

their interiors do not intersect (76).

$$\bigcup_{i=1}^{n} \mathcal{A}_i = \mathcal{M} \tag{75}$$

$$[\mathcal{A}_i/\partial\mathcal{A}_i] \cap [\mathcal{A}_l/\partial\mathcal{A}_l] = \varnothing, \; i \neq l, \; \forall i, l = 1..n \tag{76}$$

The two improvement-based solution methodologies in the following sections are based on the vertex iterations. Since these vertices are moved independently, it is possible that the edges intersecting at these vertices crosses each other, more frequently at the initial iterations. This means that the service regions are intersecting which is infeasible according to (76). Since crossing edges do not represent feasible allocation decisions, we need to perform handling (feasibility recovery) procedures for each specific event. In this section we will exemplify five such procedures. Note that there are in total nine such procedures and the remaining four are excluded for the sake of brevity. We start with the most straightforward event type and then gradually move towards more difficult vertex events.

1. A Border Vertex Exiting the Market Boundary

This vertex event is illustrated in Figure 5.8. The border vertex v_6 is initially on the market boundary at $x = M$ and if it moves in the negative y-axis direction such that it exits the lower-right corner of the market region. The figure on the right of Figure 5.8 illustrates this iteration. Note that the motion of the edge e_{14}, edge connecting v_2 and v_6, is shown as a dashed line. As part of the improvement algorithm, this iteration is allowed and the resulting final allocation solution state, shown on the right in Figure 5.8, is evaluated. In this event type one type of border vertex (v_6) is replaced with another type of border vertex (v_7').

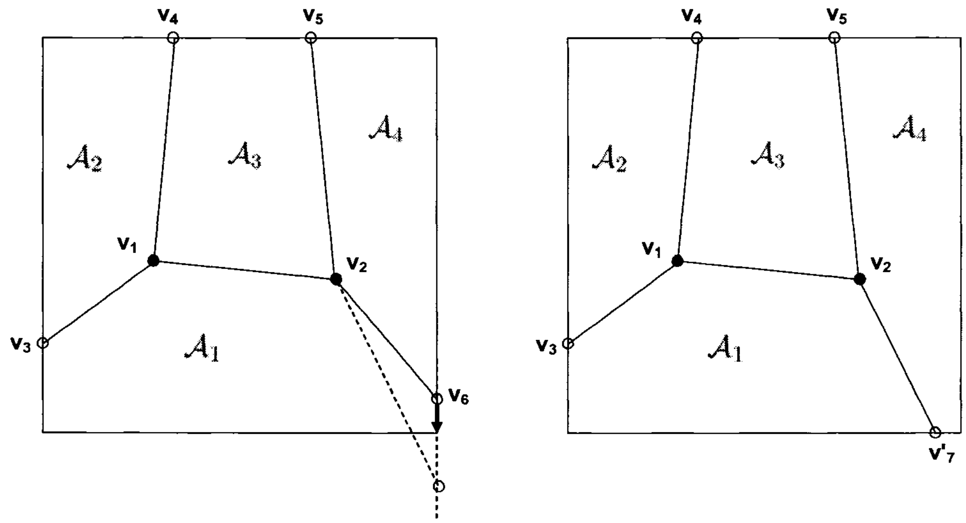

Figure 5.8: Illustration of the vertex event where a border vertex exits the market boundary.

2. Two Border Vertices Crossing

This vertex event involving two border-vertices is illustrated in Figure 5.9. The two border vertices v_4 and v_5 iterates towards and passes over each other as shown on the left. The affected edges are e_{23} and e_{34} and their motion due to this iteration are shown as dashed lines. As part of the improvement algorithm, this iteration is allowed and the resulting final allocation solution state, shown on the right in Figure 5.9, is evaluated. Observe that one internal (v_8') and one border (v_7') vertex are created. Whereas the coordinates of v_8' is determined by the intersection of the two edges, the x-coordinate of v_7' is not obvious. In our approach we take the median location implied by the iteration of the two original border vertices.

3. An internal vertex exits market boundary

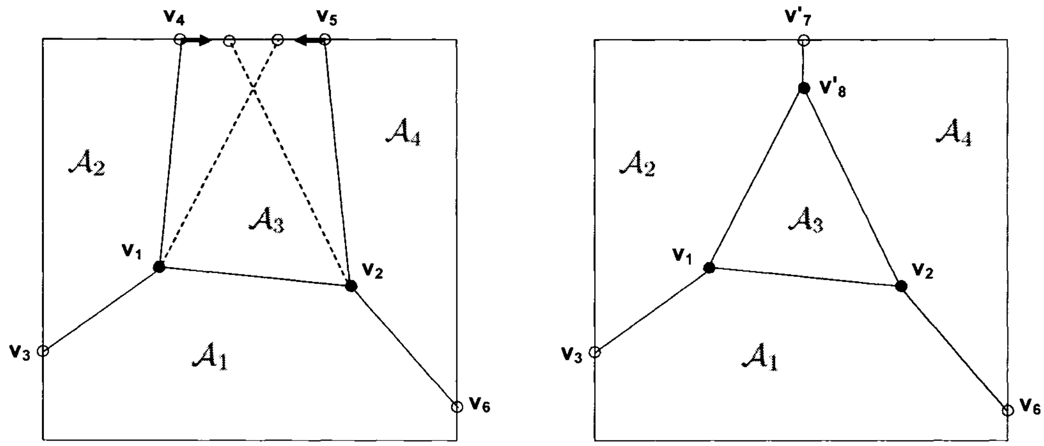

Figure 5.9: Illustration of the vertex event where two border vertices crosses over each other.

This vertex event, due to market-boundary crossing internal-vertex, is illustrated in Figure 5.10. The direction of the exiting internal vertex, v_1, is indicated with an arrow together with the filled-circle outside as the final position attempted with this iteration. The affected edges are e_{23}, e_{13} and e_{12} and their motion due to this iteration are shown as dashed lines. As part of the improvement algorithm, this iteration is allowed and the resulting final allocation solution state, shown on the right in Figure 5.10, is evaluated. When the internal vertex leaves the list of vertices, two border vertices v_7' and v_8' are created. In addition the edge e_{12} is excluded from the list of edges since it falls outside the market boundary.

4. An internal vertex crosses an edge

This vertex event, involving one internal vertex (v_1) and an edge (e_{34}), is illustrated in Figure 5.11. An internal vertex v_1 is iterated in the direction indicated with an arrow to the filled-circle within the allocation polygon $\mathcal{A}_4$. In this motion, v_1 crosses the edge e_{34} and the other affected edges are e_{23}, e_{12}

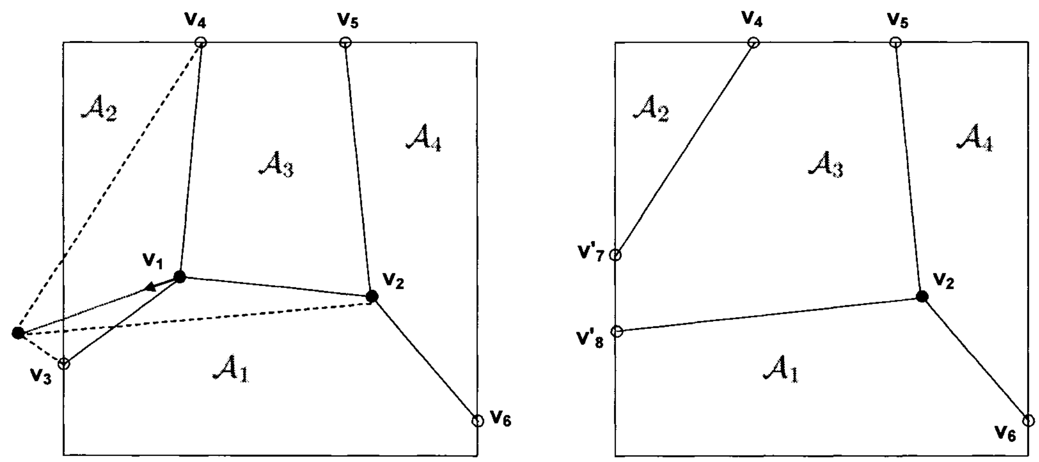

Figure 5.10: Illustration of the vertex event where an internal-vertex exits the market boundary.

and e_{13} whose motion due to this iteration are shown as dashed lines. As part of the improvement algorithm, this iteration is allowed and the resulting final allocation solution state, shown on the right in Figure 5.11, is evaluated. As a result of this iteration step, two internal vertices, v_7' and v_8', are created and original internal vertices v_1 and v_2 have disappeared. Further, the edge e_{13} disappears since its two vertices (v_1, v_2) have been excluded from the vertex list and e_{14} is relocated due to the disappearance of v_2.

5. Two internal vertices crisscrosses

This vertex event involving two internal vertices, v_1 and v_2, is illustrated in Figure 5.12. Internal vertices v_1 and v_2 iterate in the direction indicated with arrows to the filled-circles within the allocation polygon $\mathcal{A}_4$ and $\mathcal{A}_2$, respectively. In this motion, the two edges of v_1 and v_2, other than the e_{13} connecting them, crisscrosses each other shown as dashed lines. As part of the improvement algorithm, this iteration is allowed and the resulting final allocation solution state, shown on the right in Figure 5.12, is evaluated. As

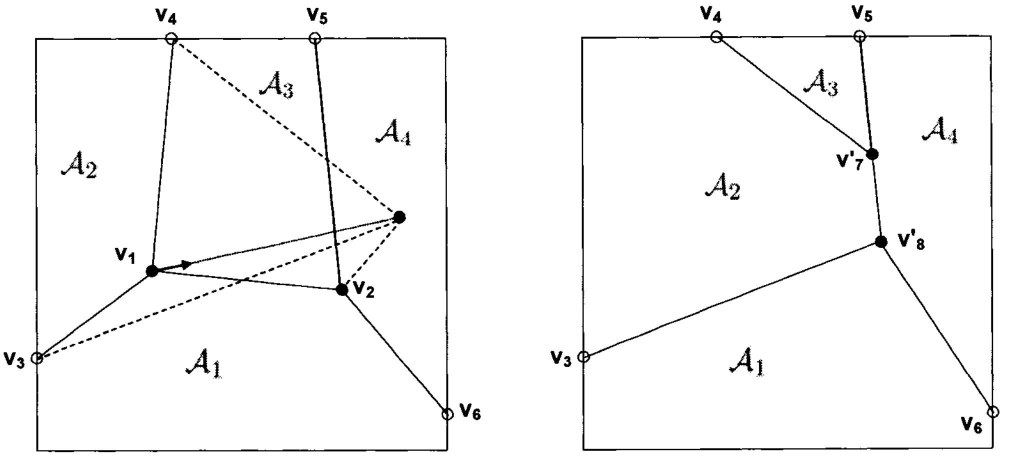

Figure 5.11: Illustration of the vertex event where an internal-vertex crosses an edge.

a result of this iteration step, two internal vertices, v_7' and v_8', are created and original internal vertices v_1 and v_2 have disappeared. Further, the edge e_{13} is replaced with the new edge e_{24} between v_7' and v_8'.

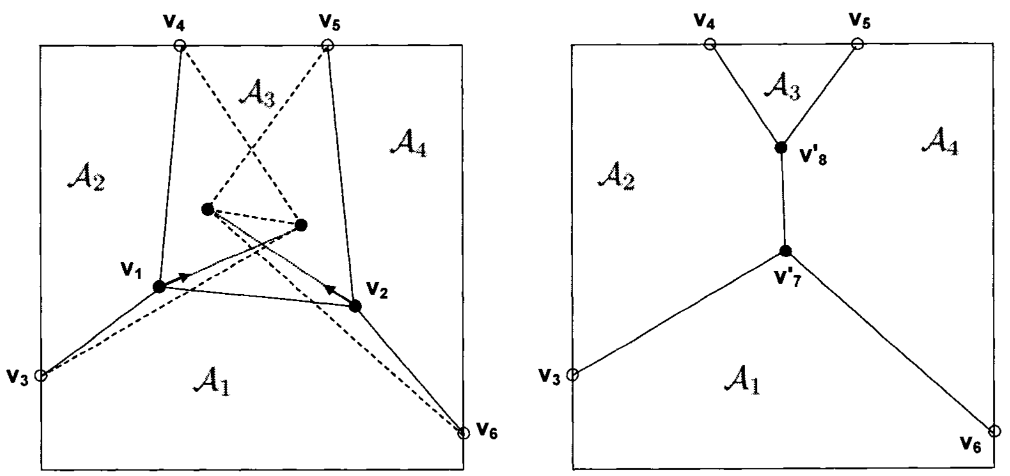

Figure 5.12: Illustration of the vertex event two internal-vertices criss-crosses each other.

5.4.3 Steepest-Descent Improvement Algorithm

In this section, we describe the steepest-descent improvement algorithm for the LAM-AVS based on the vertex-representation described in the previous section. The main structure of the algorithm is similar to the steepest-descent method for allocation decisions presented in the 2-facility case. However, as described in earlier, we adopt a vertex-iteration based improvement solution approach for the n-facility case. This change of iteration variables changes the algorithm in several ways which will be described next.

One main difference is the addition of the *vertex-event handling procedures*, some of which are described in the preceding section. Whenever the vertices are iterated in an improving direction, the structure of the allocation solution changes thus we implement an intermediate vertex-event handling procedure to generate a feasible solution. These vertex-event handling procedures aim to recover the feasibility of the allocation solution as a result of independent vertex iterations. Note that allocation decisions in the form of *non-convex sets* would still satisfy the two feasibility conditions (75) and (76). Therefore, when the vertex-event handling procedures generate non-convex allocation decision, as in Figure 5.13, it is accepted as long as the objective function value improves. Note that this type of non-convex allocation decisions are not possible in the case of voronoi diagrams, since it can be shown that nearest-neighbour property disallows such non-convex sets. However, as it will be shown in Proposition 5.4., our vertex-iteration based approach converges to a solution where the allocation decisions are in the form of convex sets. In other words, whereas we allow non-convex sets, our final allocation solution is in the form of convex sets.

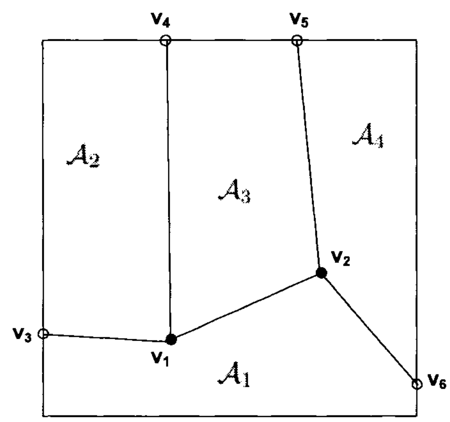

Figure 5.13: Illustration of a feasible allocation solution where one of the allocation polygons are non-convex.

The second difference is the complexity of n-facility steepest-descent algorithm compared to the one in 2-facility case. In order to mediate the increased computational burden, we adopt three modifications: Variable resolution numerical integration, Newton's method based on finite-differences for solving Fermat-Weber problems, and Inexact line-search procedure using Armijo's rule. We now briefly describe these approaches.

Since it is impossible to perform implicit integrations in the calculation of objective function, locations, and derivatives, we utilize numerical integration methods. The method we use is based on the **Gaussian quadratures**, which chooses the locations of the function evaluations as well as the weighing coefficients for them (Press et al. 1988). It thus has the advantage over the Newton-Cotes formulas (i.e. trapezoidal, Simpson's rule) in choosing the optimal locations. Since at the early stages of the improvement algorithm the gains, i.e. improvement over the previous iteration's objective function value, is larger, we can afford less numerical accuracy. However, as the solution approaches to a local minima, the gain diminishes thus the numerical accuracy becomes crucial for quality of the iteration directions. Therefore, before each line search in the improving direction, we perform a resolution update based on the incumbent gain and the current level of numerical accuracy. With this approach, initial iterations' numerical calculations are significantly less.

When the number of facilities is large, then the computation time spent in solving the single facility location problem, Fermat-Weber problem, becomes a significant overhead. Eventhough Weiszfeld's method provides a rather simple way of calculating the iterate which converges to the optimal median location, the number of iterations is significant. Based on our empirical observations, number of necessary Weiszfeld's iterations increase especially with the non-convex shapes. Since Weiszfeld's iterate is based on the first-order condition, for a faster convergence, we adopt Newton's method. Since Newton's method requires calculation of both the gradient and the Hessian, we utilize difference approximations for both of them. The forward difference approximation of first order derivatives require *two* additional function calls for any bivariate function, $f(x, y)$. For second-order derivatives it requires additional *three* function calls over first order derivatives. Thus in total we call the $f(\cdot)$ *six* times versus the central difference approximations which require more than *double* that number. These forward difference approximations used in the solving Fermat-Weber problem is summarized as below.

$$
\begin{aligned}
\frac{\partial f}{\partial x} &= \frac{f(x+h, y) - f(x, y)}{h} \\
\frac{\partial^2 f}{\partial x^2} &= \frac{f(x+2h, y) - 2f(x+h, y) + f(x, y)}{h^2} \\
\frac{\partial^2 f}{\partial x \partial y} &= \frac{f(x+h, y+h) - f(x+h, y) - f(x, y+h) + f(x, y)}{h^2}
\end{aligned}
$$

Final modification is the inexact line search method which is based on the Armijo's rule. For more details on this approach, we refer the reader to (Bazaraa et al. 1993). Herein we adopt the implementation described in this reference. We first choose a fixed step length, α_0, then we compare it with the desired level of descent based on the Armijo's test parameter and first-order approximation of the objective function. If α_0 leads to a higher objective function value than the first order approximation, then we decide on a update parameter which reduces the step length at every line search iteration, i.e. $0 < \phi < 1$. If it leads to a lower function value, then we choose an amplifying update parameter , i.e. $1 < \phi < 2$. In the former case, line search is continued until first-order approximation of the objective function is lower. In the later case, line search is continued until first-order approximation of the objective function exceeds the exact one.

We now provide a convergence result for our improvement-based algorithms which are based on the vertex-iterations. This proposition guarantees that our allocation solutions coincide with that of location-based methods, i.e. all allocation polygons are convex-sets.

Proposition 5.4.

Starting from an initial solution, a vertex-iteration based improvement approach which uses the gradient information in (73) and (74) converges to an allocation solution where all allocation polygons are convex sets.

Proof.

We will briefly establish this result with a contradiction proof and use the Figure (5.13) as an illustrative example. Assume that our algorithm has con-

verged to a feasible, but non-convex solution as shown in Figure (5.13). Our convergence condition is achieved when the gradients in (73) and (74) are zero for all six vertices. Let's assume that for v_1 and v_2, these gradients are zero. Furthermore, x_1^*, x_2^* and x_3^* are such that, the gradient for v_3 and v_4 are also zero, i.e. e_{12} and e_{23} are bisectors. Then it can be shown that the bisector of x_1^* and x_3^* would be different than the edge e_{13} connecting v_1 and v_2. Therefore the gradient term for e_{13}, which would be zero if and only if e_{13} is a bisector, is not zero. For the vertex v_1, gradients (73) and (74) are therefore nonzero (terms for e_{12} and e_{23} are zero and term for e_{13} is nonzero). This is a contradiction to our initial assumption thus our algorithm would not converge to an allocation solution where there is at least one allocation polygon which is a non-convex set. . ■

In what follows, we first provide the abridged version of the steepest-descent algorithm based on the vertex-iterations for the L_2 and L_2^2. Then we illustrate it on an example.

Steepest-Descent Improvement Algorithm based on Vertex Iterations:

Step 1. Define and Initialize the model parameters and variables

j : index for optimality iterations (i.e. $j^* = \{j | \, \epsilon_{COST} \geq \frac{|TC^{j+1} - TC^j|}{|TC^j|}\}$

ϵ_{COST} : epsilon parameter for optimality stopping decision

α^j : step length for line search iterations at the j^{th} iteration

$\mathcal{A}_i^j$: set of allocation decisions (polygons and not necessarily convex) at iteration j

$(\mathbf{x}_i^*)^j = (x_i^*, y_i^*)^j$: optimal locations corresponding to $\mathcal{A}_{i=1,2,..,n}^j$

$\mathcal{V}^j$: set of vertices formed by the intersection of three or more edges in $\mathcal{A}^j_{i=1,2,..,n}$ at iteration j

$\mathcal{E}^j$: set of edges defining $\mathcal{A}^j_{i=1,2,..,n}$ at iteration j

M : market boundary parameter (assuming a square-shaped market region)

ε : test parameter for Armijo's line search rule

ϕ^j : step size update parameter for Armijo's line search rule at iteration j

ϵ_{NUMINT} : parameter sets for 1- and 2-dimensional Gaussian quadrature integration schemes

Set $j = 0$

Step 2. Initialization: Allocate the service regions and optimally locate facilities

Step 2.1. Generate the an initial set of voronoi-generator points $p_{i=1,2,...,n}$

Step 2.2. Generate the voronoi-diagram of the $p_{i=1,2,...,n}$ and identify the following sets:

· Allocation polygons $\mathcal{A}^j_{i=1,2,...,n}$

· Set of vertices $\mathcal{V}^j$

· Set of edges $\mathcal{E}^j$

Step 2.3. Find the optimal locations and calculate the total cost. Starting with centroidal locations, employ Newton's method based on the forward difference approximations.

$$\mathbf{x}^*_i = (x^*_i, y^*_i)^j := \arg\min_{(x_i,y_i)} \left(\int_{\mathcal{A}_i} d_p(\mathbf{x}_i, \mathbf{x}) D(\mathbf{x}) d\mathbf{x}\right)$$
$$TC^j = \sum_{i=1,2,...n} \int_{\mathcal{A}_i} d_p(\mathbf{x}_i, \mathbf{x}) D(\mathbf{x}) d\mathbf{x}$$

Step 3. Improve the current solution with vertex iterations

Do While ($\frac{|TC^{j+1} - TC^j|}{|TC^j|} \geq \epsilon_{COST}$):

$j = j + 1$

Step 3.1. Update the numerical resolution

If $\frac{|TC^{j+1}-TC^{j}|}{|TC^{j}|}$ is less than the numerical resolution, ϵ_{NUMINT}, then increase the numerical integration resolution.

Step 3.2. Calculate the partial gradients

For each vertex $v_k \in \mathcal{V}^j$, repeat the next three steps for all edges e_{ij} intersecting at v_k:

1- Find partial derivative of TC with respect to single-dimensional allocation decisions $(A_i^x(y), A_i^y(x))$

$$\left(\frac{\partial TC}{\partial A_i^x(y)} - \frac{\partial TC}{\partial A_i^y(x)}\right) = [d_p(\mathbf{x}_i, \mathbf{x}) - d_p(\mathbf{x}_j, \mathbf{x})]\, D(\mathbf{x}) \qquad \forall e_{ij} \in E_k$$

2- Find partial derivative of TC with respect to the Slope and Intercept parameters of the edge e_{ij}

Slope (Pure rotation): $\frac{dTC}{da_{ij}} = \int_{y \in Y_{BR}} \left(\frac{\partial TC}{\partial A_i^x(y)} - \frac{\partial TC}{\partial A_i^y(x)}\right) \frac{\partial A^x(y)}{\partial a_{ij}} dy$

Intercept (Translation): $\frac{dTC}{db_{ij}} = \int_{y \in Y_{BR}} \left(\frac{\partial TC}{\partial A_i^x(y)} - \frac{\partial TC}{\partial A_i^y(x)}\right) \frac{\partial A^x(y)}{\partial b_{ij}} dy$

where $\frac{\partial A_i^x(y)}{\partial a_{ij}} = -\frac{A_i^x(y)}{a_{ij}}$ and $\frac{\partial A_i^x(y)}{\partial b_{ij}} = -\frac{1}{a_{ij}}$

3- Find partial derivative of TC with respect to v_k^x and v_k^y, i.e. $\frac{\partial TC}{\partial v_k^x}$ and $\frac{\partial TC}{\partial v_k^y}$

Using the derivative aggregation formulae in (73) and (74) to obtain $\mathbf{d}_{v_k}$

Normalize the gradients $\mathbf{d}_{v_k}$ to obtain gradient vectors, $\mathbf{d}_{v_k}^{norm}$, for $\forall v_k \in \mathcal{V}^j$

Step 3.4. Perform a line search for step size α^j

· Choose ϕ^j, step size update parameter for Armijo's line search rule

· Repeat following until Armijo's line test is satisfied (Set $t = 0$)

Set $t := t + 1$ and $\alpha^j := \alpha_0 \left(\phi^j\right)^t$

Update the vertex coordinates

$v_k' := v_k + \alpha^j \mathbf{d}_{v_k}^{norm}$

Perform Vertex-Event Handling Procedures and update $\left(\mathcal{A}_i^j\right)', (\mathcal{V}^j)'$ and $(\mathcal{E}^j)'$

Repeat **Step 2.3** using $\left(\mathcal{A}_i^j\right)', (\mathcal{V}^j)'$ and $(\mathcal{E}^j)'$

· Set $\left(\alpha^j\right)^* = \alpha_0 \left(\phi^j\right)^t$

Step 3.3. Update the allocation decisions

Update the vertex coordinates

$v_k' := v_k + \alpha^j \mathbf{d}_{v_k}^{norm}$

Perform Vertex-Event Handling Procedures and update $\mathcal{A}_i^j, \mathcal{V}^j$ and $\mathcal{E}^j$

Repeat **Step 2.3** using $\mathcal{A}_i^j, \mathcal{V}^j$ and $\mathcal{E}^j$

Return **Step 3.**

Step 4. Terminate with the solution $\mathcal{A}_i^j$ and $\mathbf{x}_i^*$

Example 5.1: Steepest-Descent Improvement Algorithm based on Vertex Iterations - Euclidean-metric (L_2) Case

We now illustrate the above algorithm on an example. In this example, we have a square-shaped market region $\mathcal{M}=\{(x,y)|x \in (0,100) \text{ and } y \in (0,100)\}$, i.e. $M = 100$. The demand density function, $D(\mathbf{x})$ is a highly nonlinear demand function shown in Figure 5.14. The distance measure is based on the Euclidean-metric (L_2).

In our example illustration, we have six service regions and the starting solution is illustrated in the Figure 5.15. Note that the facilities are at the optimal locations within their respective service regions. More importantly, the allocation lines are not equidistant.

ITERATION 1

Step 1. Define and Initialize the model parameters and variables

$\epsilon_{COST} = 1 \times 10^{-6}$

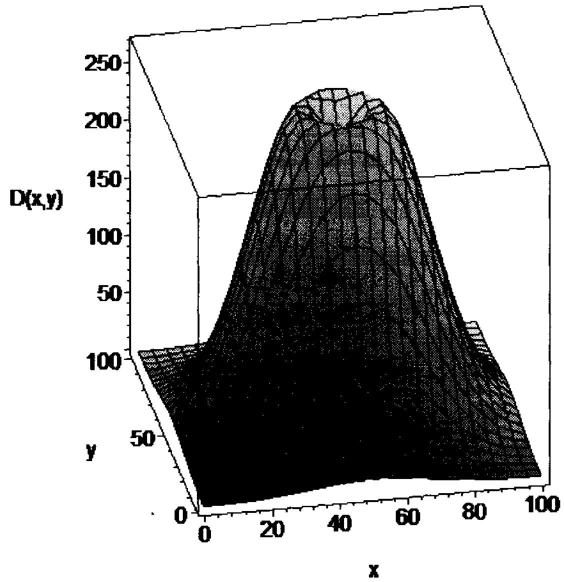

Figure 5.14: Demand density function of the example for steepest-descent algorithm on vertex iterations (Example 5.1).

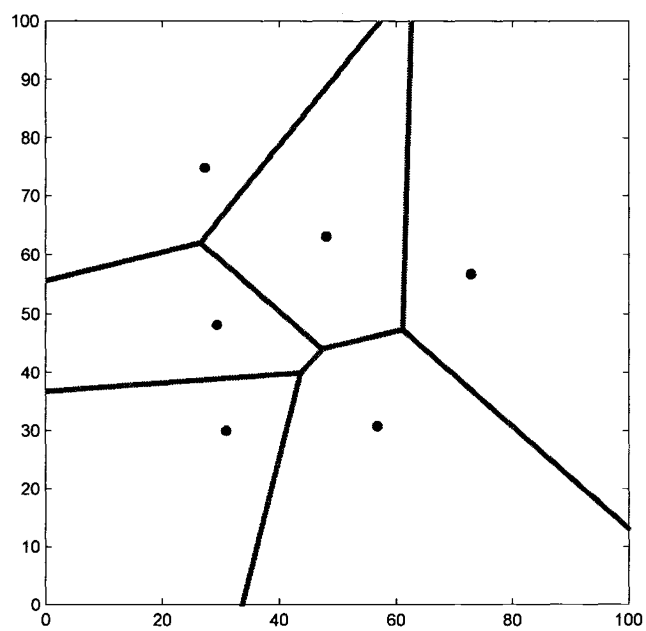

Figure 5.15: Starting solution for the example implementation of the steepest-descent algorithm based on vertex iterations (Example 5.1).

$\varepsilon = 0.3$, $\phi^j = 2$ (for this implementation ϕ is either 2 or 1/2, namely the step length is either doubled or halved)

ϵ_{NUMINT} : numerical accuracy for the integration is set at 10×10^{-4} . Note that this is the guaranteed accuracy, on the average it results in 10×10^{-5}. For example, total demand for the demand density shown in figure is $8,500,000.0$ and $\epsilon_{NUMINT} = 10 \times 10^{-4}$ results in $8,500,768.0$ which is 9.0×10^{-5} error level.

Set initial step length $\alpha_0 = 1.0$.

Step 2. Initialization: Allocate the service regions and optimally locate facilities

Initial points for the generation of the voronoi diagram are as follows.

$p_{i=1\ldots6} = \{(30,46),(80,60),(43,61),(22,78),(31,32),(51,27)\}$

Allocation polygons $\mathcal{A}_i^{j=0}$, set of vertices $\mathcal{V}^{j=0}$, set of edges $\mathcal{E}^{j=0}$ are shown in Figure 5.15.

Optimal locations are follows which are also shown in the Figure 5.15.

$$x^*_{i=1\ldots6} = \left\{ \begin{array}{c} (29.326,48.132),(72.806,56.746),(48.063,63.120), \\ (27.264,74.753),(30.948,30.044),(56.745,30.851) \end{array} \right\}$$

$TC = 105,013,252.10$

Step 3. Improve the current solution with vertex iterations

Do While $(\frac{|TC^{j+1}-TC^j|}{|TC^j|} \geq \epsilon_{COST})$:

$j = 1$

Step 3.1. Update the numerical resolution

Since it is first iteration, numerical accuracy is not a constraint.

Step 3.2. Calculate the partial gradients

We will illustrate the calculation of the partial gradients for a single vertex

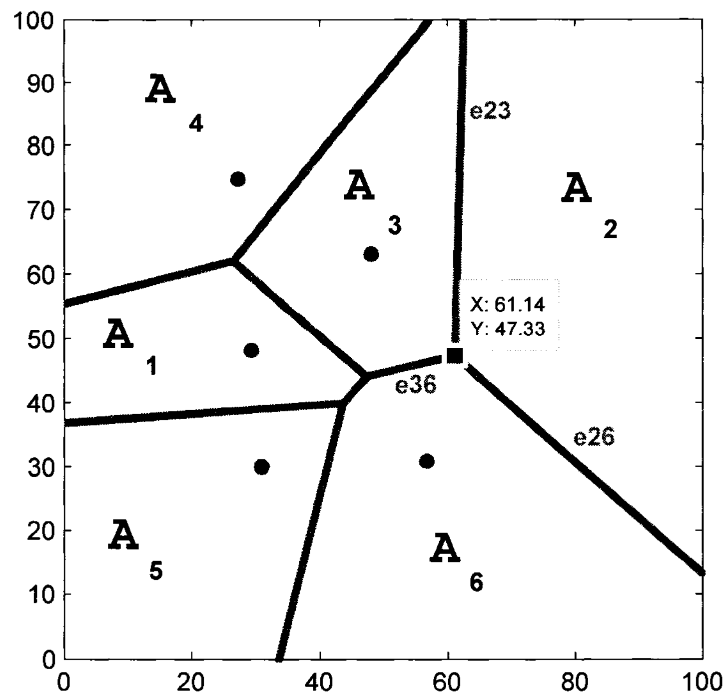

Figure 5.16: Illustration for the calculation of partial derivatives for an interior vertex (Example 5.1).

then provide the gradient results for the remaining vertices. Figure 5.16 illustrates this vertex (v_k) which is a non-degenerate vertex and is the intersection of three edges e_{36}, e_{23} and e_{26}.

For the vertex $v_k = (61.14, 47.33)$, partial derivative of TC with respect to the Slope (a) and Intercept (b) parameters of the edges are calculated as follows:

Edge e_{23} :

$A_3^x(y) = \frac{y-b_{23}}{a_{23}}$, where $a_{23} = 37.0$ and $b_{23} = -2215.0$

Slope: $\frac{dTC}{da_{23}} = \int_{y=47.33}^{y=100} [d_p(\mathbf{x}_3^*, \mathbf{x}) - d_p(\mathbf{x}_2^*, \mathbf{x})] \left(-\frac{y-b_{23}}{(a_{23})^2}\right) D(A_3^x(y), y)dy$

$= -97,209.4$

Intercept: $\frac{dTC}{db_{23}} = \int_{y=47.33}^{y=100} [d_p(\mathbf{x}_3^*, \mathbf{x}) - d_p(\mathbf{x}_2^*, \mathbf{x})] \left(\frac{-1}{a_{23}}\right) D(A_3^x(y), y)dy$

$= -1,613.2$

Edge e_{26} :

$A_6^x(y) = \frac{y-b_{26}}{a_{26}}$, where $a_{26} = -0.8787$ and $b_{26} = 101.0606$

Slope: $\frac{dTC}{da_{26}} = \int_{y=13.18}^{y=47.33} [d_p(\mathbf{x}_6^*, \mathbf{x}) - d_p(\mathbf{x}_2^*, \mathbf{x})] \left(-\frac{y-b_{26}}{(a_{26})^2}\right) D(A_6^x(y), y)dy$

$= -385,488.9$

Intercept: $\frac{dTC}{db_{26}} = \int_{y=13.18}^{y=47.33} [d_p(\mathbf{x}_6^*, \mathbf{x}) - d_p(\mathbf{x}_2^*, \mathbf{x})] \left(\frac{-1}{a_{26}}\right) D(A_6^x(y), y)dy$

$= -47,517.6$

Edge e_{36} :

$A_3^x(y) = \frac{y-b_{36}}{a_{36}}$, where $a_{36} = 0.235294$ and $b_{36} = 32.941$

Slope: $\frac{dTC}{da_{36}} = \int_{y=44.085}^{y=47.33} [d_p(\mathbf{x}_3^*, \mathbf{x}) - d_p(\mathbf{x}_6^*, \mathbf{x})] \left(-\frac{y-b_{36}}{(a_{36})^2}\right) D(A_6^x(y), y)dy$

$= -6,337,940.7$

Intercept: $\frac{dTC}{db_{36}} = \int_{y=44.085}^{y=47.33} [d_p(\mathbf{x}_3^*, \mathbf{x}) - d_p(\mathbf{x}_6^*, \mathbf{x})] \left(\frac{-1}{a_{36}}\right) D(A_6^x(y), y)dy$

$= -116,118.7$

3- Find the partial derivatives of TC with respect to v_k^x and v_k^y, i.e. $\frac{\partial TC}{\partial v_k^x}$ and $\frac{\partial TC}{\partial v_k^y}$, using the derivative aggregation formulae in (73) and (74).

$\frac{\partial TC}{\partial v_k^x} = \left(\frac{\partial TC}{\partial a_{23}}\frac{\partial a_{23}}{\partial v_k^x} + \frac{\partial TC}{\partial b_{23}}\frac{\partial b_{23}}{\partial v_k^x}\right) + \left(\frac{\partial TC}{\partial a_{26}}\frac{\partial a_{26}}{\partial v_k^x} + \frac{\partial TC}{\partial b_{26}}\frac{\partial b_{26}}{\partial v_k^x}\right) + \left(\frac{\partial TC}{\partial a_{36}}\frac{\partial a_{36}}{\partial v_k^x} + \frac{\partial TC}{\partial b_{36}}\frac{\partial b_{36}}{\partial v_k^x}\right) =$ $90,543.234$

$\frac{\partial TC}{\partial v_k^y} = \left(\frac{\partial TC}{\partial a_{23}}\frac{\partial a_{23}}{\partial v_k^y} + \frac{\partial TC}{\partial b_{23}}\frac{\partial b_{23}}{\partial v_k^y}\right) + \left(\frac{\partial TC}{\partial a_{26}}\frac{\partial a_{26}}{\partial v_k^y} + \frac{\partial TC}{\partial b_{26}}\frac{\partial b_{26}}{\partial v_k^y}\right) + \left(\frac{\partial TC}{\partial a_{36}}\frac{\partial a_{36}}{\partial v_k^y} + \frac{\partial TC}{\partial b_{36}}\frac{\partial b_{36}}{\partial v_k^y}\right) =$ $-86,564.81$

where

$\frac{\partial a_{23}}{\partial v_k^x} = 25.99$, $\frac{\partial b_{23}}{\partial v_k^x} = -1626.2$, $\frac{\partial a_{23}}{\partial v_k^y} = -0.70246$, $\frac{\partial b_{23}}{\partial v_k^y} = 43.951$.

$\frac{\partial a_{26}}{\partial v_k^x} = -0.02261653$, $\frac{\partial b_{26}}{\partial v_k^x} = 2.261653$, $\frac{\partial a_{26}}{\partial v_k^y} = -0.0257361$, $\frac{\partial b_{26}}{\partial v_k^y} = 2.573605$

$\frac{\partial a_{36}}{\partial v_k^x} = -0.01707379$, $\frac{\partial b_{36}}{\partial v_k^x} = 0.80866586$, $\frac{\partial a_{36}}{\partial v_k^y} = 0.07256362$, $\frac{\partial b_{36}}{\partial v_k^y} = -3.43683$

Normalize the gradients $\mathbf{d}_{v_k}$ to obtain gradient vectors, $\mathbf{d}_{v_k}^{norm}$, for $\forall v_k \in \mathcal{V}^j$

Step 3.4. Perform a line search for step size α^j

$\alpha_0 = 1$ lead to a better objective value than the first-order approximation, thus set $\phi^{j=1} = 2$

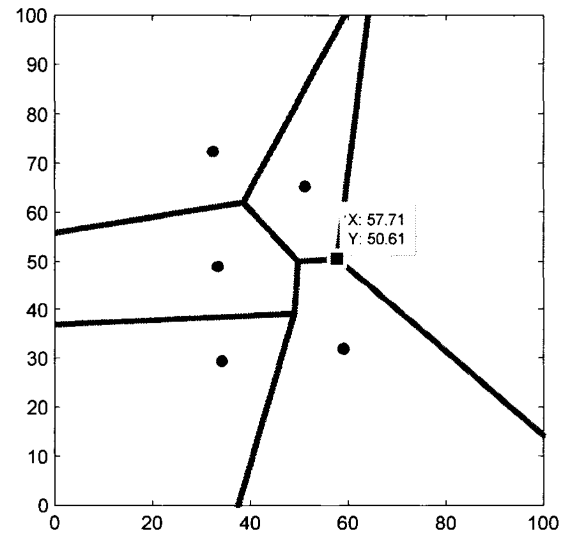

Figure 5.17: Solution at the end of first iteration for the example implementation of the Steepest-descent algorithm based on vertex iterations (Example 5.1).

Perform Vertex-Event Handling Procedures and update $\left(\mathcal{A}_i^j\right)', (\mathcal{V}^j)'$ and $(\mathcal{E}^j)'$

Repeat **Step 2.3** using $\left(\mathcal{A}_i^j\right)', (\mathcal{V}^j)'$ and $(\mathcal{E}^j)'$

Line search takes $t = 5$ iterations and terminates with $\left(\alpha^{j=1}\right)^* = \alpha_0 (2)^5 = 16$

Step 3.3. Update the allocation decisions

Update the vertex coordinates

$v_k' := v_k + \alpha^j \mathbf{d}_{v_k}^{norm}$

Perform Vertex-Event Handling Procedures and update $\mathcal{A}_i^j, \mathcal{V}^j$ and $\mathcal{E}^j$

Repeat **Step 2.3** using $\mathcal{A}_i^j, \mathcal{V}^j$ and $\mathcal{E}^j$

End of ITERATION 1

Solution at the end of this first iteration is illustrated in Figure 5.17, together with the vertex (v_k) analyzed in the first iteration.

Table 5.1. presents the iteration results for this example. Note that we have employed an adaptive numerical accuracy procedure; thus, when the integration accuracy is not satisfactory, it is increased. The solution converges to a local optima with $TC = 99,121,760.4$. The final solution is illustrated in the Figure 5.18.

Iter. No	TC	Ave. Line Search Iter.
0	10,501,325.21	-
1	10,235,151.91	3.2
5	10,002,782.78	2.8
10	9,930,857.70	1.8
15	9,919,435.55	1.6
20	9,914,812.62	2.2
25	9,913,619.03	2.4
30	9,912,843.88	2.8
35	9,912,265.18	3.4
40	9,912,182.40	-

Table 5.1. Steepest-descent improvement algorithm's iteration results for the Euclidean-metric (L_2) example (Example 5.1).

5.4.4 Conjugate-Gradient Improvement Algorithm

In this section we describe the conjugate gradient method as an alternative to the steepest-descent improvement approach for the vertex-iterations. The conjugate gradient methods' performance lies in between that of steepest-descent and Newton's method and requires calculation of only the first order derivatives. This simplicity of the conjugate gradient method is the reason why we have chosen it over Newton's methods.

It is also worthwhile to explain why the conjugate gradient method is preferable over the Quasi-Newton methods which also requires first order derivative information and iteratively approximates the Hessian. As described

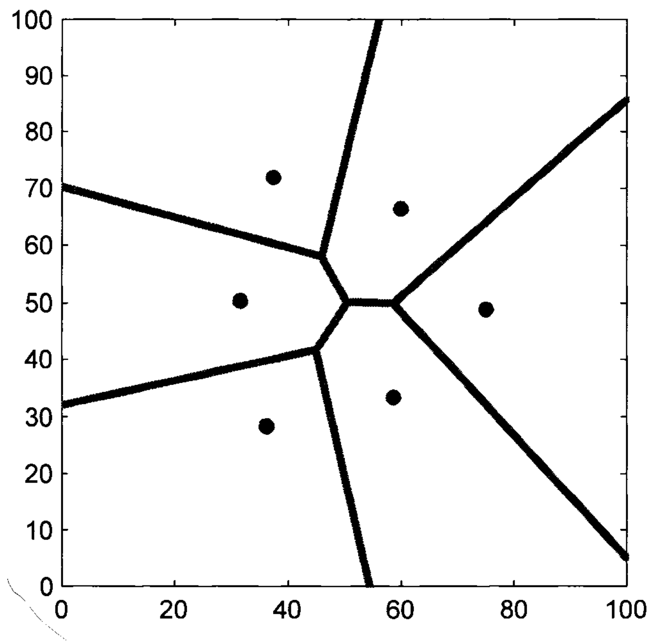

Figure 5.18: Final solution at the end of 40^{th} iteration for the example implementation of the Steepest-descent algorithm based on vertex iterations (Example 5.1).

in the preceding section, vertex iterations results in special vertex events which are handled according to certain rules. The impact of these vertex-event handling procedures is that the variable space of the problem is constantly being updated, such that certain vertices are removed from the variable space and others are included in the variable space. Quasi-Newton methods approximates the Hessian iteratively by using the update from the previous iterations. Therefore the inclusion and exclusion of variables would destroy the accuracy of the Hessian matrix for the incumbent vertex set.[11] Since, in general, the performance of the conjugate gradient methods lies between those of steepest-descent and quasi-newton methods, we choose conjugate gradient method. In our implementation, we use the Polak-Ribiere update which is known to perform better than its alternatives (Bazaraa et al. 1993).

The only difference between the steepest-descent algorithm presented in the previous section and the conjugate gradient is the **Step 3.2.**, where the search direction is calculated based on the following formula.

$$\mathbf{d}_v^{j+1} = -\mathbf{g}^{j+1} + \beta^j \mathbf{d}_v^j \tag{77}$$

$$\beta^j = \frac{\left(\mathbf{g}^{j+1} - \mathbf{g}^j\right)^T \mathbf{g}^{j+1}}{\left(\mathbf{g}^j\right)^T \mathbf{g}^j} \tag{78}$$

where

$\mathbf{d}_v^j$: direction vector of the vertices at iteration j

$\mathbf{g}^{j+1}$: gradient vector for TC with respect to vertices, i.e. $\mathbf{g}^{j+1} = \nabla_{v_k} TC$

As part of the convergence requirement of the conjugate gradient method, (78) is reset at every $|\mathcal{V}^{j_L}|$ iterations, namely the cardinality of the vertex set

[11]However, it is also possible to track the updates on the approximate Hessian and filter according to the incumbent vertex set, but this type of selective reconstruction of the approximate Hessian is left as a future research direction.

at the last reset (Bazaraa et al. 1993). The initial value of $\mathbf{d}_v^j$ and $\mathbf{g}^j$ are set at:

$$\mathbf{g}^j = \nabla_{v_k} TC \quad \text{and} \quad \mathbf{d}_v^j = -\mathbf{g}^j \text{ for } \forall \operatorname{mod}(j, |\mathcal{V}^j|) = 0$$

Note that since some of the vertices are leaving and entering the vertex variable set, we need to account for their conjugate directions accordingly. This is done through a filtering procedure at Step 3.2., where the exiting vertices' rows are removed from $\mathbf{g}^j$ and, for the new vertices, the corresponding rows in $\mathbf{g}^j$ are set at 0.

Since the remaining algorithmic steps of conjugate gradient approach is similar to that of steepest-descent, we omit formal description of the conjugate gradient improvement algorithm based on vertex iterations. However, to demonstrate the differences in the convergence rates of the two methods, we implement the conjugate gradient method to the example in the previous section. Table 5.2. compares the performances of the two methods. From this table, it is apparent that conjugate-gradient method has a superior convergence rate near the solution compared to the steepest descent. Note that its initial performance is not as good as the steepest-descent, which is attributable to the fact that vertex-events degrades the conjugacy of the directions generated. As the solution approaches to the local minima, there is less of such vertex events thus the benefit of the conjugate gradient method is more pronounced.

	Stepest Descent		Conjugate Gradient	
Iter. No	TC	Ave. Line Search Iter.	TC	Ave. Line Search Iter.
0	105,013,252.10	-	105,013,252.10	-
1	102,351,519.10	3.2	102,351,519.10	3.4
5	100,027,827.80	2.8	100,188,610.80	2.8
10	99,308,577.00	1.8	99,427,093.70	1.8
15	99,194,355.50	1.6	99,144,761.00	1.8
20	99,148,126.20	2.2	99,128,679.40	1.6
25	99,136,190.30	2.4	99,121,825.10	-
30	99,128,438.80	2.8		
35	99,122,651.80	3.4		
40	99,121,824.00	-		

Table 5.2. Comparison of Steepest-descent and Conjugate-gradient methods for the Euclidean-metric (L_2) example (Example 5.1).

5.5 Conclusions

In this chapter, we have provided a solution framework for the planar n-facility location-allocation problems in the allocation variable space when the distance measure is based on Euclidean-metric. Our solution framework, in some part, constitutes as an extension to the one developed for the 2-facility case. The framework developed here is based on vertex representation of the allocation decisions for a special class of distance measures which are shape invariant, such as Euclidean-metric based distance measures.

The vertex-iteration based representation allows the application of shape preserving transformation concepts which form the basis of the improvement approaches developed in the 2-facility case. While the allocation decision representation of the 2-facility case for Euclidean-metric based measures is in the form of a straight line, in the n-facility case allocation decisions are represented in more complex structures. For this complication, we utilized the vertex and edge representation of the voronoi diagrams. Using the results developed for the allocation decisions in the 2-facility case, we adopted a vertex-based representation and establish the analytical relations between these vertices and the allocation decisions, i.e. edges. This approach allows designing solution methodologies for the n-facility location-allocation problem in the allocation variable space. Since this approach is based on the independent movement of the vertices, the resulting solution can lead to an infeasible allocation solution. Therefore, we additionally developed a set of event handling procedures which not only recovers the feasibility but also regards the directional improvement in the allocation decisions. Due to this trade-off between the recovery of solution feasibility and allowing as much directional improvement as possible, the resulting solution can be non-convex allocation decisions. This is in contrast with the solution techniques in the location space where allocation decisions are always convex sets. Even though intermediate steps of our approach can result in non-convex solutions, the final solution we converge to coincides with that of location-based methods. Based on the vertex iterations concept, two improvement-based solution procedures are developed: Steepest-descent and Conjugate gradient method. Whereas, the steepest descent method performs satisfactorily except for the slow rate of convergence near local solutions, the conjugate gradient method displays a faster convergence rate as it approaches to a local solution.

Chapter 6

Computational Experiments: Euclidean-Metric

6.1 Introduction

In this chapter we report on our computational results. This computational study is conducted to investigate the following relations:

i) Investigate the effect of the number of facilities with different demand density functions

ii) Investigate the effect of the transportation cost parameter on the number of facilities

iii) Understand how demand density affects the objective as well as the solutions

The model used is the Euclidean-metric planar n-facility location-allocation problem in the allocation variable space (LAM-AVS), which is repeated below from Chapter 5. Accordingly, the solution method used is the steepest-descent improvement algorithm presented in the same chapter.

LAM- Allocation Variable Space (LAM-AVS):

$$\min_{\substack{A_i^y(x), A_i^x(y) \\ i=1,2,\ldots,n}} TC(A_i^y(x), A_i^x(y), \mathbf{x}_i^*, i = 1, 2, ..., n)$$

$$s.t.$$

$$\begin{aligned} (B_i^x(y) + A_i^x(y), y) &= (x, B_i^y(x) + A_i^y(x)) \qquad for \ \ (x, y) \in \partial \mathcal{A}_i \\ \mathbf{x}_i^* &= \arg\min_{(\mathbf{x}_i)} \int_{\mathcal{A}_i} d_{L_2}(\mathbf{x}_i, \mathbf{x}) D(\mathbf{x}) d\mathbf{x} \qquad \text{for } i = 1, 2, ..., n \end{aligned}$$

where

$$TC(A_i^y(x), A_i^x(y), \mathbf{x}_i^*, i = 1, 2, ..., n) = \sum_{i=1}^{n} \begin{pmatrix} F + f + a \int\limits_{\mathcal{A}_i} D(\mathbf{x})d\mathbf{x} \\ +c \int\limits_{\mathcal{A}_i} d_{L_2}(\mathbf{x}_i^*, \mathbf{x})D(\mathbf{x})d\mathbf{x} \end{pmatrix}$$

Each service region's cost is composed of fixed facility cost (F), fixed-charge linear capacity acquisition cost ($f + a \int\limits_{\mathcal{A}_i} D(\mathbf{x})d\mathbf{x}$), and the transportation cost ($c \int\limits_{\mathcal{A}_i} d_{L_2}(\mathbf{x}_i^*, \mathbf{x})D(\mathbf{x})d\mathbf{x}$). In all problems, we assume a square-shaped market region $\mathcal{M}$, i.e. $\mathcal{M}=\{(x, y)|x \in (0, 100)$, and $y \in (0, 100)\}$. Note that, when total demand volume ($\int\limits_{\mathcal{A}_i} D(\mathbf{x})d\mathbf{x}$) is constant in two different problems, then the capacity acquisition cost of each problem is identical.[12] In this chapter, we will refer to ($\sum_{i=1..n} \int\limits_{\mathcal{A}_i} d_{L_2}(\mathbf{x}_i^*, \mathbf{x})D(\mathbf{x})d\mathbf{x}$) as the *total traveled distance.*

We developed the steepest-descent algorithm code in Matlab and performed runs for a randomly generated problem set. We have restricted the number of iterations for the steepest descent to 200. However, the majority of the runs converged in less iterations than the 200 limit. For example, for $n = 5$ and LD-1, Figure 6.1 presents the solutions for all 27 iterations where convergence to a local solution is achieved.

In Section 6.2, we describe the experimental design in more detail. Section 6.3 presents results for the computational complexity and run-time performance of the steepest descent algorithm for different demand density functions as well as number of facilities. Section 6.4 and Section 6.5 present results for linear and nonlinear demand density functions.

[12] This is because we are assuming linear capacity acquisition cost with which total capacity acquisition cost is same for $\sum_{i=1}^{n} \left(a \int\limits_{\mathcal{A}_i} D(\mathbf{x})d\mathbf{x} \right)$ and $\left(a \int\limits_{\mathcal{M}} D(\mathbf{x})d\mathbf{x} \right)$.

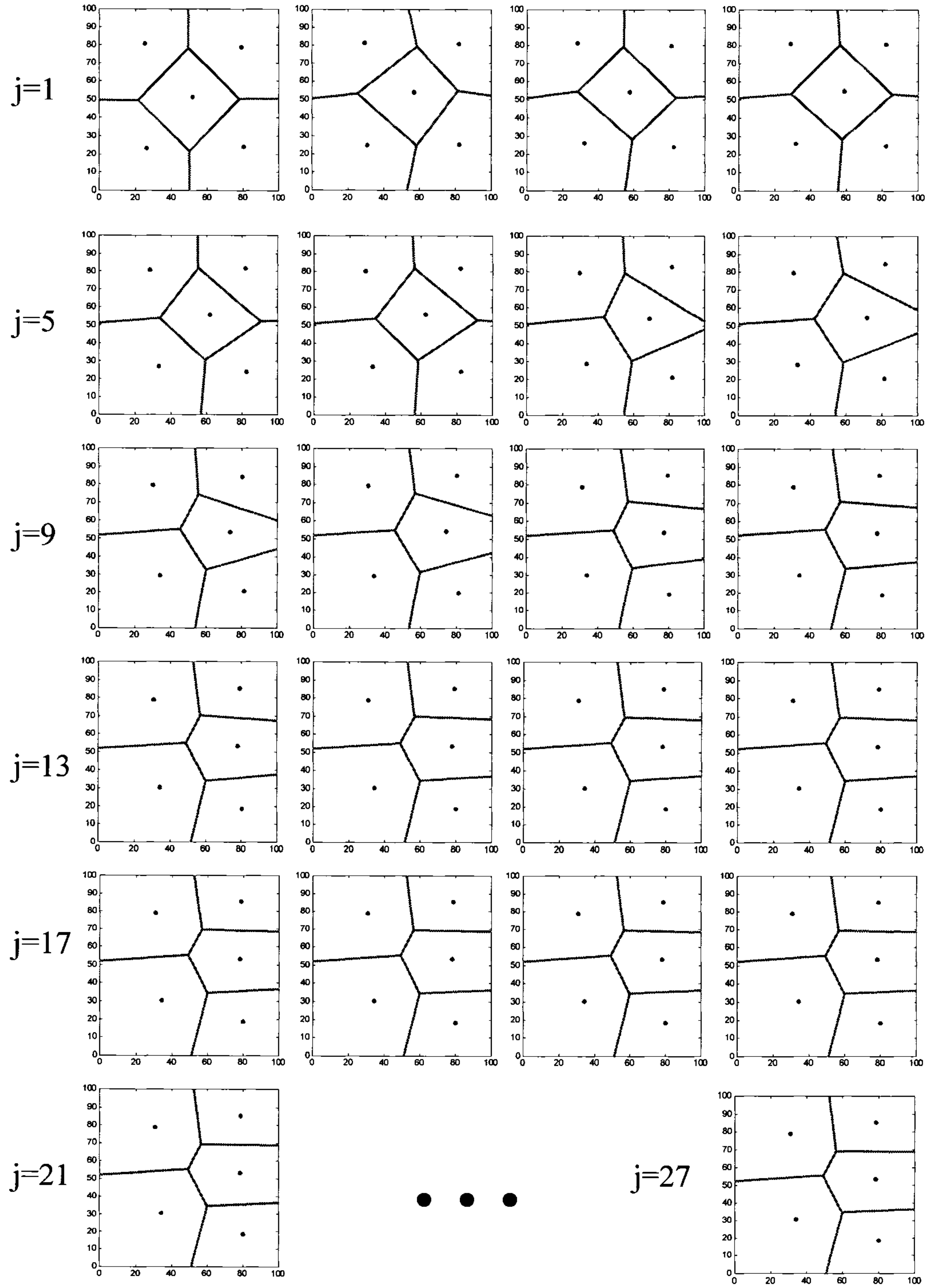

Figure 6.1: Complete iteration results for $n = 5$ facility with the demand density function $D(\mathbf{x}) = 100 + 10x + 5y$, i.e. LD-1 in Table 6.1.

6.2 Experimental Design

In our experimental design, we consider two main categories of demand density functions: Linear and Nonlinear demand density functions. For each category, we have chosen six different demand density functions, which are illustrated in Table 6.1. and Table 6.2. The first column in both tables includes the reference labels used hereafter. The second column is self explanatory, whereas the third column is described in the next paragraphs. Furthermore, we consider five facility combinations for each demand density, i.e. n = 3, 5, 8, 10, and 15 facility cases. In total, 60 (=12 demand functions × 5) problems are solved. In order to increase the confidence of the results, we performed ten runs for each problem starting with different initial solutions. These starting points are chosen randomly. Hence, the total number of experimental runs on which we base our results is 600. Note that we are reporting on the solutions which are best among ten random starts.

In order to compare the solutions without need for any adjustment, we setup a demand density function such that the total demand volume is constant for all demand types.

$$\int_{\mathcal{M}} D(\mathbf{x})d\mathbf{x} = 8,500,00.0 \text{ units for } \forall D(\mathbf{x}) \text{ in Table 6.1. and Table 6.2.}$$

With this setup, the variable component of the capacity acquisition cost is same for all problems, i.e. $8.5\times10^6\,(a)$. However, total fixed costs $(\sum_{i=1..n}(F+f))$ and the transportation costs are still different for problems with different demand function and/or number of facilities. Note that for a given demand pattern (shape), the total volume of demand does not change the optimal allocation solution.[13]

Table 6.1. illustrates linear demand functions. First column reference $LD - \#$ stands for "*Linear Demand*". Last column of the Table 6.1., $\frac{D_{\max}}{D_{\min}}$, measures the ratio of maximum demand density level ($D_{\max}$) in the market region $\mathcal{M}$ to the minimum demand density level ($D_{\min}$). This measure, though not an exact measure of the variance, allows us to judge the rate of change of the demand density within the market region. Note that demand functions (LD-1, LD-2, LD-3) represent rapidly varying demand, whereas (LD-4, LD-5, LD-6) represent slowly varying demand. Herein, this labeling (i.e. slowly and rapidly varying) is comparative. Furthermore, LD-1 and LD-3 (or LD-4 and LD-5) differs from each other in their single-dimensional demand density change, which will be discussed in the following sections.

Demand Density Reference	D(x,y)=	Dmax/Dmin
LD-1	$100+10x+5y$	16.00
LD-2	$100+7.5x+7.5y$	16.00
LD-3	$100+\frac{100x}{7}+\frac{5y}{7}$	16.00
LD-4	$600+\frac{10x}{3}+\frac{5y}{3}$	1.83
LD-5	$600+2.5x+2.5y$	1.83
LD-6	$600+\frac{100x}{21}+\frac{5y}{21}$	1.83

Table 6.1: List of linear demand density functions

Table 6.2. illustrates the nonlinear demand density functions. Again the

[13] Consider two demand functions $LD-1$ and $2\times D(x) = 200+20x+10y$. The solution (i.e. allocation and location decisions) would be same for these two functions, but the objective function would be different.

last column allows us to compare the rate of change in the demand density across the market region. Last two demand functions, NLD-5 and NLD-6 are Newling type demand density functions which represent the distribution of urban population in major cities (Newling, 1969). Note that $\frac{D_{\max}}{D_{\min}}$ ratio of LD-6 is notably higher than others. All six nonlinear demand cases are illustrated in Figure 6.2.[14]

Demand Density Reference	D(x,y)=	Dmax/Dmin
NLD-1	$950-\frac{3(x-50)^2}{50}-\frac{3(y-50)^2}{50}$	1.46
NLD-2	$1200-\frac{21(x-50)^2}{100}-\frac{21(y-50)^2}{100}$	8.00
NLD-3	$750+\frac{3(x-50)^2}{50}+\frac{3(y-50)^2}{50}$	1.40
NLD-4	$300+\frac{33(x-50)^2}{100}+\frac{33(y-50)^2}{100}$	6.50
NLD-5	$\frac{854115}{1372}\mathbf{e}^{\left(-\left(\frac{((x-50.)^2+(y-50.)^2)^{0.5}}{1000}-0.05\right)((x-50.)^2+(y-50.)^2)^{0.5}\right)}$	8.08
NLD-6	$2000\mathbf{e}^{\left(-\left(\frac{2579((x-50.)^2+(y-50.)^2)^{0.5}}{1188439}-0.05\right)((x-50.)^2+(y-50.)^2)^{0.5}\right)}$	2003.88

Table 6.2: List of nonlinear demand density functions

[14]Due to the ease of their visualization, we have excluded the similar plots for linear demand density cases.

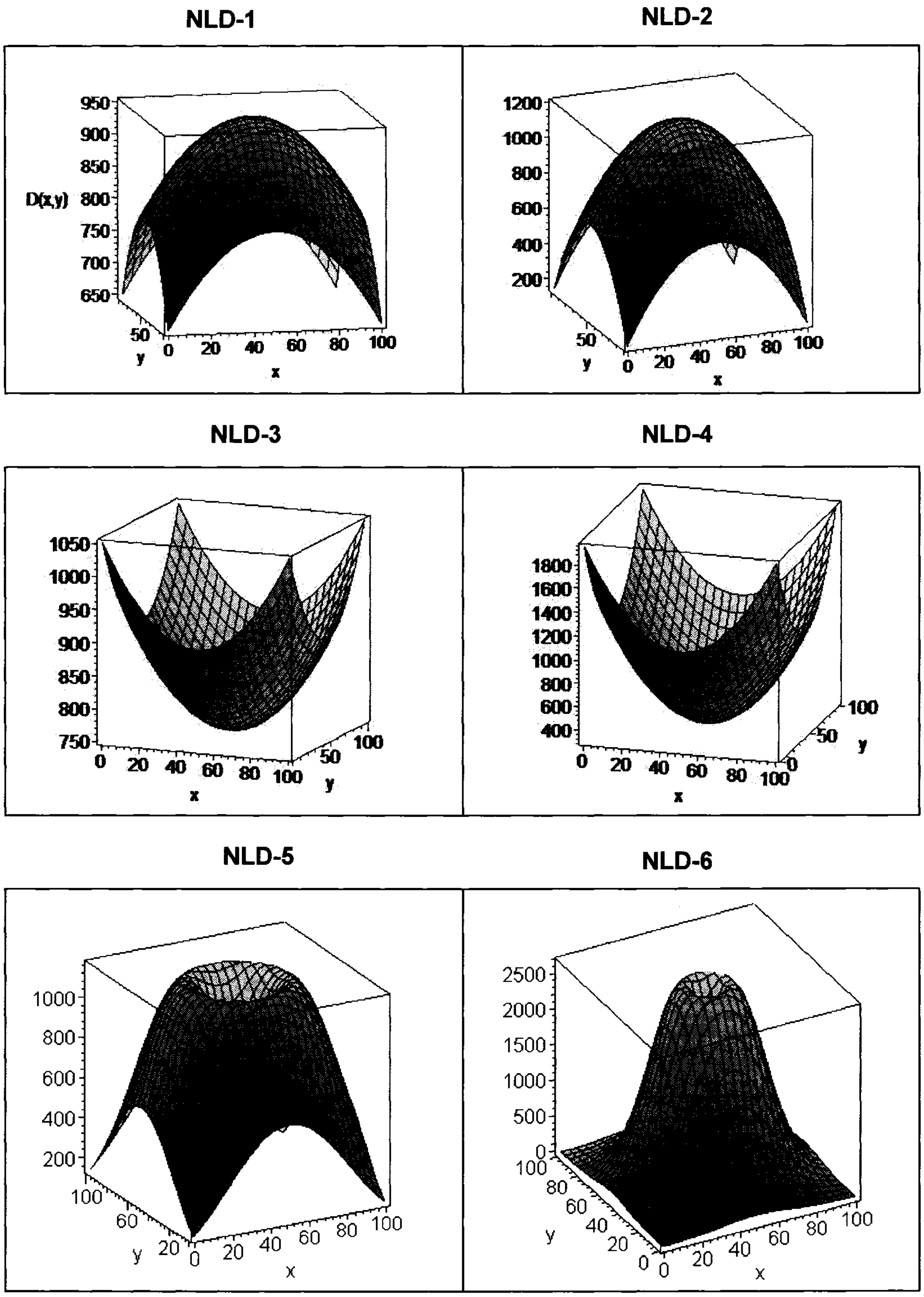

Figure 6.2: Six nonlinear demand density functions used in the experimental study.

6.3 Computational Complexity

In this section we report on the computational runtime performance of the steepest-descent algorithm applied to LAM-AVS. In particular, we present some results on the following:

a) Effect of the number of facilities on the number of iterations for convergence

b) Effect of the demand density pattern on the number of iterations for convergence

Note that (a) and (b) are both dependent on the starting solution.

Before discussing (a) and (b), we first illustrate the effect of *numerical tolerance* on the computational time. In Table 6.3., we illustrate the CPU time for one iteration of the problem with $n = 15$ and NLD-6.[15] Numerical tolerance, in second column, represents the tolerance parameter set in the steepest-descent algorithm for numerical integration calculations. This is the minimum tolerance level required from the numerical integration; however, actual accuracy (last column) is significantly better. Last column represents the deviation of total demand in column five from the exact value $8,500,000.0$. Therefore in our implementations, we start with an initial tolerance of 1×10^{-2} and reduce it gradually as per need basis (Chapter 5). Note that the majority of the time, approximately 94%, is spent in solving single-facility location problems (Fermat-Weber solution). Fermat-Weber problems are solved with a difference approximation based Newton's method to a minimum precision level of 1×10^{-4}.

[15]Results are from the Matlab Profiler on a P4 2.8MHz and 1GB RAM PC. Note that times reported also includes overhead times, i.e. actual run times are lesser.

Case No	Numerical Integral Tolerance	Total Time (seconds)	Fermat-Weber Solution Time (seconds)	Total Demand	Actual Accuracy
1	1.0E-07	307.391	294.141	8,500,000.03	3.3E-09
2	1.0E-06	97.969	93.328	8,500,000.20	2.3E-08
3	1.0E-05	44.703	42.578	8,500,012.30	1.4E-06
4	1.0E-04	21.516	20.281	8,500,097.20	1.1E-05
5	1.0E-03	11.141	10.484	8,500,768.00	9.0E-05
6	1.0E-02	6.5	6.016	8,506,139.00	7.2E-04

Table 6.3: Effect of numerical tolerance on the computational performance (one iteration of the problem with $n = 15$ and NLD-6)

In order to investigate the effect of the demand density on the solution performance we have experimented with three demand types: linear, quadratic and fourth-order convex polynomial. These demand functions ($LD2$, $NLD-a$, $NLD-b$) are as below and illustrated in Figure 6.3.

$$\begin{aligned}
LD2 &: \quad D(x,y) = 100 + 7.5x + 7.5y \\
NLD-a &: \quad D(x,y) = 100 + \frac{9}{80}x^2 + \frac{9}{80}y^2 \\
NLD-b &: \quad D(x,y) = 100 + \frac{3}{1.6\times 10^5}x^4 + \frac{3}{1.6\times 10^5}y^4
\end{aligned}$$

Table 6.4. illustrates the number of iterations for three demand densities as well as four different number of facilities, i.e. $n = 4,\ 9,\ 16,$ and 36. The numerical tolerance parameter for each of these runs are set at 1×10^{-3}. The gap (**% Difference**) between the solution converged and the best known local solution are also illustrated in Table 6.4.[16] From the number of iterations in Table 6.4., it is apparent that as we increase the number of facilities, the number of iterations increases polynomially. However, each iteration takes longer,

[16] Best known solution is the solution when we increase the numerical integration tolerance to 1×10^{-7} and continue with the iterations until the convergence is attained with the local solution gap of 0.1, i.e. $\left(TC^k - TC^{k-1}\right) \leq 1 \times 10^{-1}$.

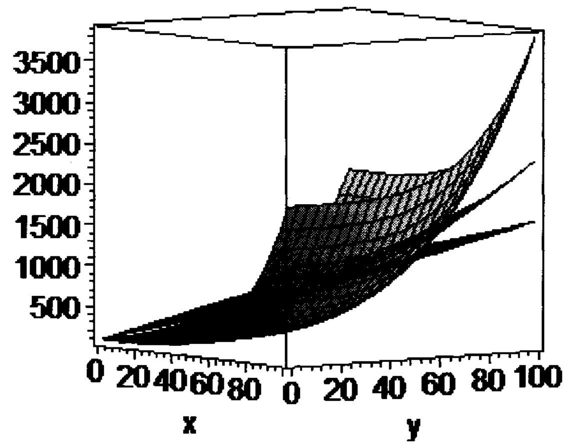

Figure 6.3: Illustration of the three demand density functions for computational comparison.

as we are solving more Fermat-Weber problems. For example, in Table 6.3 (for $n = 15$), Fermat-Weber solutions takes 10.5 seconds with 1×10^{-3} numerical accuracy. When we increase the number of facilities to $n = 36$, single-facility location solution would amount to approximately 25 seconds per iteration. Hence, with a large n, starting with a small numerical integration tolerance is computationally more attractive. Furthermore, from Table 6.4, there is a clear difference between the linear demand density ($LD2$) and the nonlinear demand density ($NLD - b$). Based on this example, we could conclude that nonlinear density functions require more descent iterations than the linear demand functions.[17]

[17]Note that at every iteration, we use numerical integration procedures which converge faster to the tolerance limit when the functions are linear than the case where they are nonlinear. However, this is the effect of linearity/nonlinearity on the time spent per iteration rather than the number of iterations for convergence to a local solution.

	ITERATIONS			% Difference		
n	**LD2**	**NLD-a**	**NLD-b**	**LD2**	**NLD-a**	**NLD-b**
4	11	8	20	0.0009%	0.0048%	0.0009%
9	13	12	20	0.0020%	0.0075%	0.0089%
16	22	25	41	0.0043%	0.0089%	0.0093%
36	51	58	85	0.0065%	0.0091%	0.0096%

Table 6.4: Number of steepest-descent iterations and percentage difference from best solution for three different demand density functions and four levels of number of facilities

Finally, note that our vertex-iteration based algorithms (both the steepest-descent and conjugate-gradient methods) have the order of complexity proportionate to the number of facilities as illustrated in Table 6.5. Note that in the table, the number of vertices and edges are assumed to be at their theoretical maximum levels (see Section 5.1.1). Furthermore, "**k**" represents the coefficient for the border vertices, and C_1 and C_2 are constants. Hence, it is apparent that each iteration of the steepest-descent has the order of complexity proportionate to the number of facilities n.

Step	**Operation**	**Complexity**
0-	Start with an initial allocation solution	-
1-	Calculate the gradient information for each edge	O(3n-6)
2-	Calculate gradient information for each vertex	O(2n-5+kn)
3-	Construct feasible allocation polygons	
a)	Iterate vertices one at a time	O(2n-5+kn)
b)	Check for vertex events	O(2n-5+kn)
c)	Apply one of the vertex event handling operations	O(9(2n-5+kn))
	Total	$O(C_1n+C_2)$

Table 6.5: Iteration steps of the steepest-descent algorithm and respective orders of complexity.

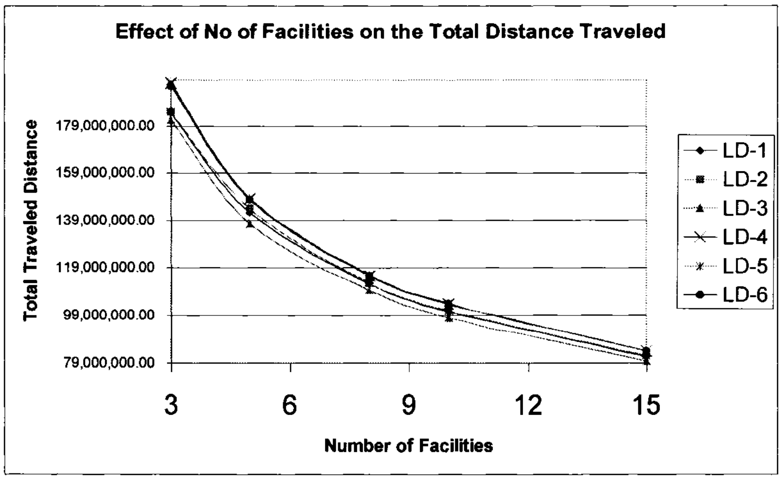

Figure 6.4: Graph illustrating the effect of number of facilities on the total distance traveled when the demand is linear.

6.4 Linear Demand

In this section, we will analyze the effect of number of facilities, transportation cost parameter, and demand density function on the objective function and allocation solutions based on the demand types listed in Table 6.1.

6.4.1 Effect of Number of Facilities

Figure 6.4 illustrates the results for five different number of facilities. One major observation is that all the six demand density functions exhibit similar diminishing returns on the total traveled distance as we increase the number of facilities. In other words, average travel distance per unit is nonlinearly increasing as we decrease the number of facilities, i.e. increase the service region sizes. This observation conforms to the findings in Chapter 3, where we showed that transportation cost is proportionate to the square of the service region size.

6.4.2 Effect of Transportation Cost

As stated before, all linear demand cases have the same demand volume. Therefore, we parametrically vary the unit transportation cost parameter (c) to compare its effect on the optimal solution characteristics such as the number of facilities. For this we have chosen three c values, i.e. $c = 0.75, 1.0$, and 1.25, to denote **low**, **average** and **high** unit transportation costs. In addition, we have specified total fixed cost, i.e. summation of the fixed facility cost (F) and capacity-acquisition fixed cost (f), to be $F + f = 10 \times 10^5$. As an example, for $n = 3$, total traveled distance for LD-1 is $184,803,765.05$ (Table 6.6.). Hence, when $c = 0.75$, we have a total cost of $(0.75)\,(184,803,765.05) + 3 \times 10 \times 10^5 = 168,602,823.79$. This can be verified in Figure 6.5.

Figure 6.5 illustrates the case $c = 0.75$, i.e. low unit transportation cost. Observe that for all demand types $n = 5$ is the best number of facilities. Figures 6.6 and 6.7 illustrate the similar results for the $c = 1.0$ and $c = 1.25$ cases. One major insight, though intuitive, is that the ideal number of facilities increases with increasing unit transportation cost.

Another interesting, but not intuitive, observation is the faster rate of change in the total cost when number of facilities is less than the ideal compared to the case when it is higher. This is due to nonlinearly increasing transportation cost in proportion to service region sizes. Also note that as we increase c, the relative importance of the transportation cost increases with respect to the fixed-cost terms. Therefore, Figure 6.7 with $c = 1.25$ illustrates this observation more distinctly than other figures. A major insight from this observation is that it is better to locate more facilities, as oppose to less, than the ideal number. We will discuss this insight in combination with another observation more in the next section.

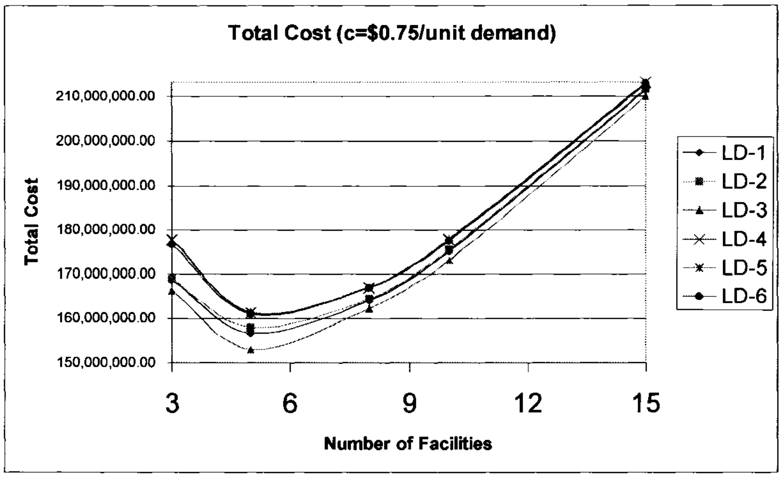

Figure 6.5: Total cost as a function of the number of facilities with transportation cost parameter $c = \$0.75/unit$ (Linear Demand).

6.4.3 Effect of Demand Density Functions

Table 6.6. presents the total traveled distance results. In our analysis of these results, we can safely ignore the fixed cost of facilities as well as the capacity acquisition costs. This is because our interest is to compare the effect of demand density function parameters on the solution for **a given number of facilities.** In what follows, we first analyze the effect of rate of change in the demand density over the market region. Later, we will consider the cases when the demand density varies differently in $x-$ and $y-$ dimensions.

Recall from Table 6.1. that (LD-4, LD-5, LD-6) represent slowly varying demand, whereas (LD-1, LD-2, LD-3) represent rapidly varying demand. With this information, Table 6.6., illustrates that slowly varying demand cases have **consistently** higher total traveled distance than the rapidly varying demand cases. Note that this observation is only valid for monotonically increasing

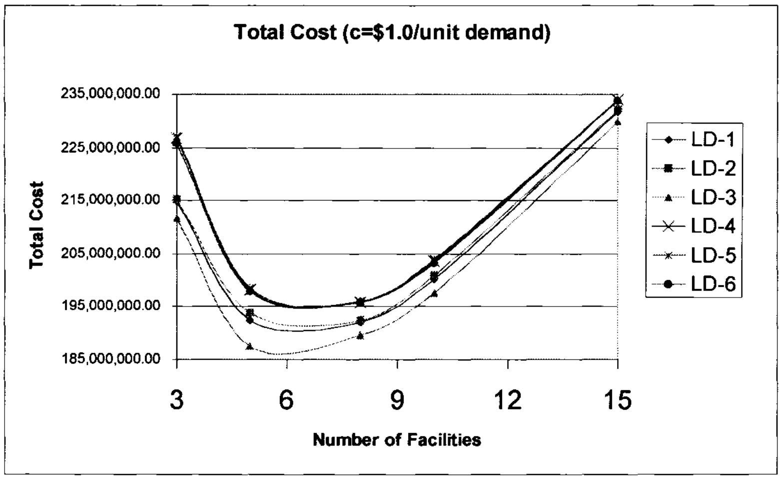

Figure 6.6: Total cost as a function of the number of facilities with transportation cost parameter $c = \$1.00/unit$ (Linear Demand).

linear demand cases.

No. Facilities	LD-1	LD-2	LD-3	LD-4	LD-5	LD-6
3	184,803,765.05	185,215,428.90	181,576,613.59	196,876,265.18	197,167,624.16	195,761,444.47
5	142,330,893.12	143,757,192.30	137,403,639.46	148,136,833.23	148,398,168.33	147,835,857.01
8	112,038,045.38	112,397,414.32	109,487,410.57	115,768,966.16	115,917,307.41	115,675,150.36
10	100,169,828.10	100,824,618.50	97,537,650.88	103,590,107.97	103,924,703.57	103,290,646.69
15	81,718,664.59	82,048,901.66	79,941,673.20	83,917,100.37	83,932,548.68	83,791,205.22

Table 6.6: Total traveled distance results with five different number of facilities when demand is linear

In order to better understand the effect of the rate of change (i.e. $\frac{D_{\max}}{D_{\min}}$) in demand density, consider Table 6.7. Second, third and fourth columns are percentage differences (of the total traveled distance) between pairs (LD-1, LD-4), (LD-2, LD-5) and (LD-3, LD-6), respectively. These pairs are chosen in order to isolate any other functional differences than the $\frac{D_{\max}}{D_{\min}}$. For example, LD-1 and LD-2 are identical in terms of ratio $\frac{v}{w}$ where v and w are the coefficients of x and y coordinates in the demand, i.e. $D(\mathbf{x}) = u + vx + wy$. In each of these pairs, first one varies more rapidly than the second one, thus

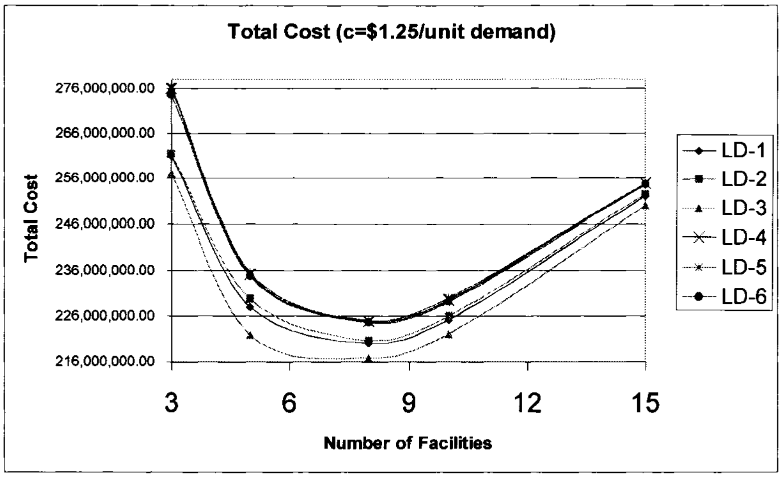

Figure 6.7: Total cost as a function of the number of facilities with transportation cost parameter $c = \$1.25/unit$ (Linear Demand).

the percentage is calculated by taking the first one as the base level. ROW_{Max} (ROW_{Min}) represents the highest(lowest) total traveled distance of the corresponding row. The last column represents the difference (between the highest and lowest) as a percentage of the lowest. To illustrate, consider $n = 3$ facilities. The lowest value in the first row is $181,576,613.59$ for LD-3 and the highest value is $197,167,624.16$ for the LD-5, and 8.59% is obtained as the percentage difference. Results of Table 6.7. illustrate that the rate of change in demand density has more impact when the number of facilities is low. One insight from this result is that when the number of facilities are small, the planner should demand higher accuracy in the demand data.

Another insight, from the results thus far, is related to the under-/over-estimating the demand variation. When we underestimate the variation, i.e. assume slowly varying, then we tend to locate more facilities since the transportation cost is a significant portion of the system cost. In contrast, when we overestimate we locate less number of facilities. Note that these errors are due to inaccurate information on the demand distribution. If we compare these two error alternatives, underestimation (slowly varying assumption) is better than overestimation (rapidly varying) since the cost of erring on the higher side of the ideal number of facilities is less than on the lower side (Figure 6.7).

No. Facilities	$\frac{(LD\text{-}4)-(LD\text{-}1)}{(LD\text{-}1)}$	$\frac{(LD\text{-}5)-(LD\text{-}2)}{(LD\text{-}2)}$	$\frac{(LD\text{-}6)-(LD\text{-}3)}{(LD\text{-}3)}$	$\frac{(ROW_{Max}-ROW_{Min})}{ROW_{Min}}$
3	6.53%	6.45%	7.81%	8.59%
5	4.08%	3.23%	7.59%	8.00%
8	3.33%	3.13%	5.65%	5.87%
10	3.41%	3.07%	5.90%	6.55%
15	2.69%	2.30%	4.82%	4.99%

Table 6.7: Effect of the rate of change in the demand density on the total distance traveled

Figure 6.8 illustrates the difference in the solutions for two demand instances, LD-5 for slowly-varying demand and LD-2 for rapidly-varying demand. Observe that the solutions are considerably different. Similar patterns hold true for other pairs, i.e. (LD-4, LD-1) and (LD-6, LD-3). Hence, we conclude that the rate of change in demand affects not only the objective function (Table 6.7.) but also the solution (Figure 6.8).

We now turn to the cases where demand varies differently in $x-$ and $y-$dimensions. Either of the triplets (LD-1, LD-2, LD-3) or (LD-4, LD-5, LD-6) can be analyzed for this type of variation. We choose the former triplet and order the demand functions as in (LD-2, LD-1, LD-3). While in the function of LD-2 x and y share the same coefficients, LD-3's coefficients have the highest difference. Figure 6.9 illustrates problems with $n = 5, 8,$ and 15 for this ordering. We note that LD-2's solution is moderately different than LD-1's,

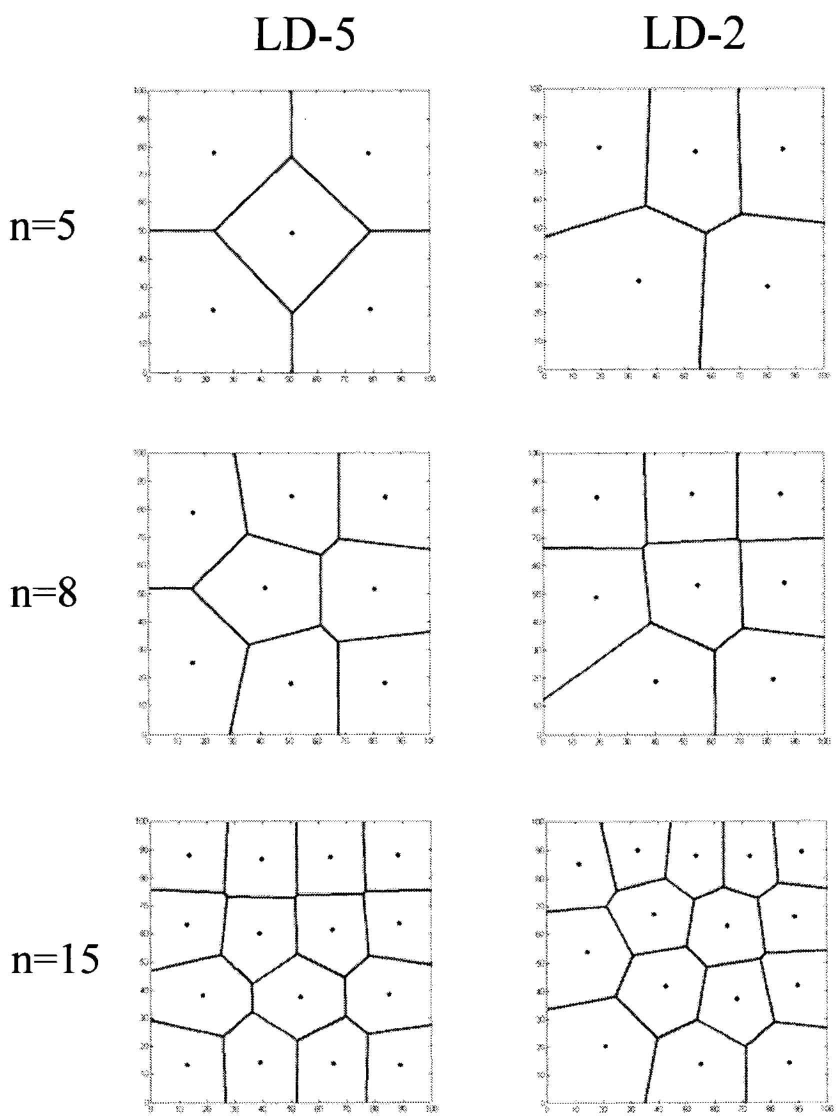

Figure 6.8: Solution to $n = 5, 8$, and 15 facility cases for demands LD-5 and LD-2.

whereas LD-1 and LD-3's difference is mute, especially with fewer facilities. This result, based on visual comparison, is also supported by the results in Table 6.6. However, note that the gap between LD1 and LD3's objectives is higher than the gap between LD1 and LD2's objectives, which is not obvious from the Figure 6.9. Nevertheless, these gaps are not comparable to the gap between slowly and rapidly varying demand cases. Hence, we could speculate that overall variation (i.e. $\frac{D_{\max}}{D_{\min}}$), is a much stronger determinant of objective differences than the axis dependent variation in demand density.

6.5 Nonlinear Demand

6.5.1 Effect of Number of Facilities

Figure 6.10 illustrates the results for five different numbers of facilities. As in the case of linear demand density, all the six cases exhibit similar diminishing returns on the total traveled distance as we increase the number of facilities. One particular difference is the case NLD-6 which not only has significantly lower total travel but also is less sensitive to the decrease in the number of facilities.

6.5.2 Effect of Transportation Cost

We parametrically vary the unit transportation cost parameter (c) to compare its effect on the optimal solution characteristics such as the number of facilities. For this we choose the same three c values, i.e. $c = 0.75, 1.0$, and 1.25, as before. Total fixed costs is same as before, i.e. $F + f = 10 \times 10^5$. Figures 6.11, 6.12 and 6.13 illustrate all three cases. Since the pattern of these graphs is similar to the linear demand case, earlier observations apply to the nonlinear

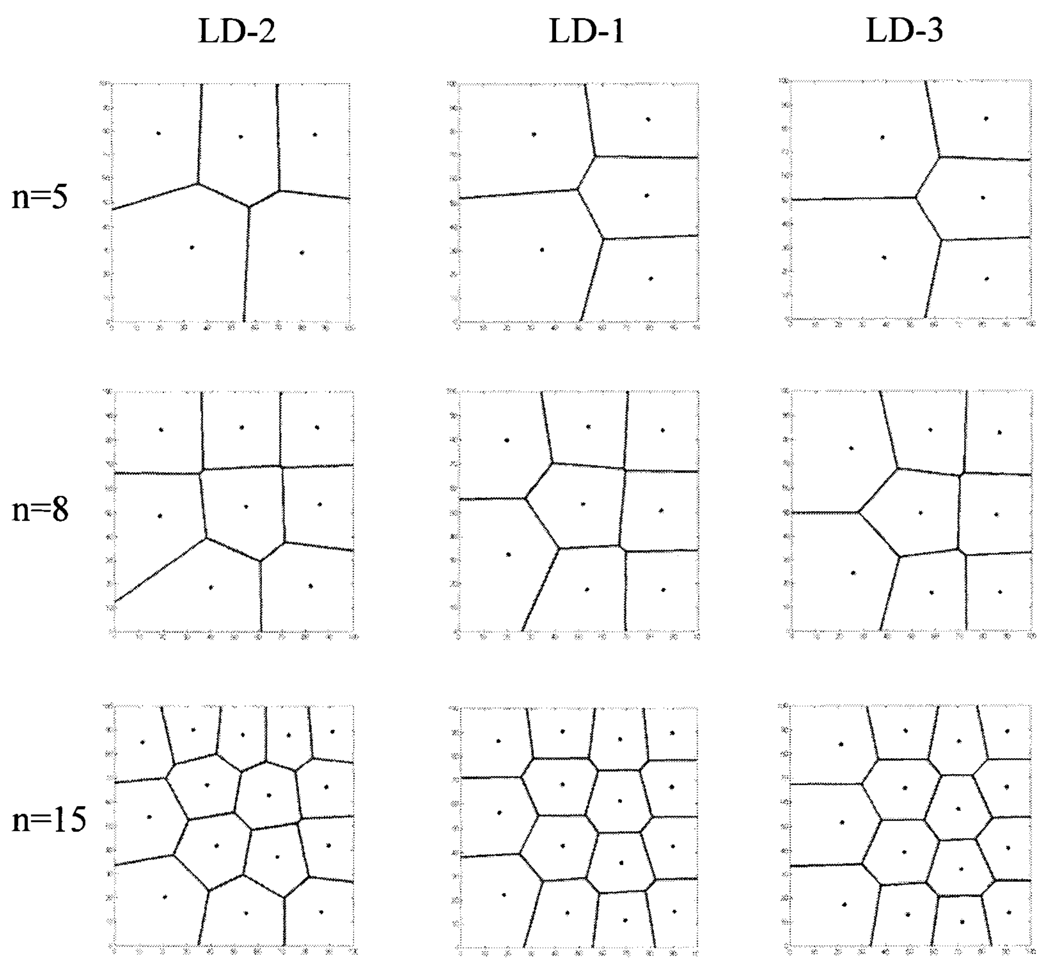

Figure 6.9: Solutions to $n = 5, 8$ and 15 facilities for demand functions LD-2, LD-1, and LD-3.

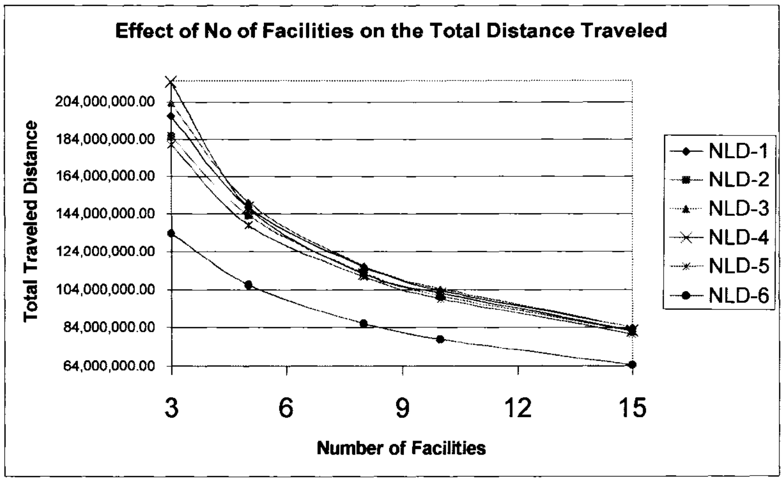

Figure 6.10: Graph illustrating the effect of number of facilities on the total distance traveled when the demand is nonlinear.

demand as well. One difference is the NLD-6, which represents a highly-varying demand scenario. Note that its ideal number of facilities is relatively insensitive to the transportation cost parameter.

6.5.3 Effect of Demand Density Function

Table 6.8. presents the total traveled distance for six nonlinear demand cases. We again compare the effect of demand density function parameters on the solution for **a given number of facilities**. In what follows, we first analyze the effect of rate of change in the demand density on the objective function. Later, we will discuss these differences from the solutions perspective using visual representations of the solutions.

Recall from Table 6.2. that NLD-1, NLD-3, and NLD-5 represent a slowly varying demand in comparison with NLD-2, NLD-4, and NLD-6, respectively. Table 6.8. illustrates that slowly varying demand cases have **consistently** (with the exception of $n = 3$ for NLD-3 and NLD-4) higher total traveled

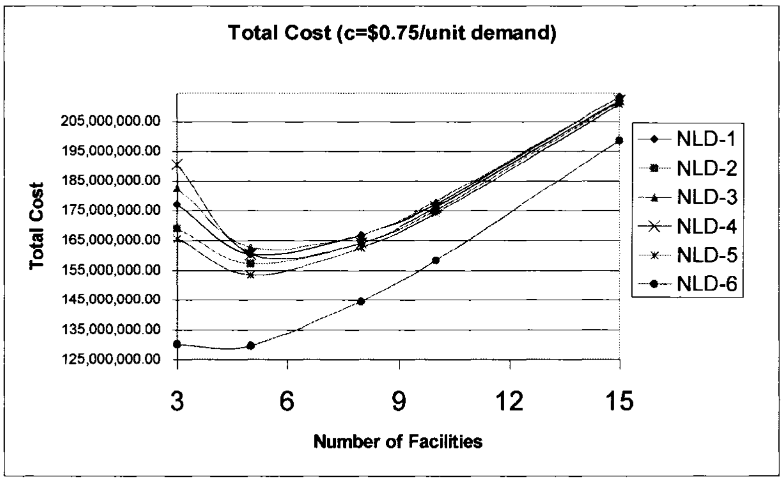

Figure 6.11: Total cost as a function of the number of facilities with transportation cost parameter $c = \$0.75/unit$.

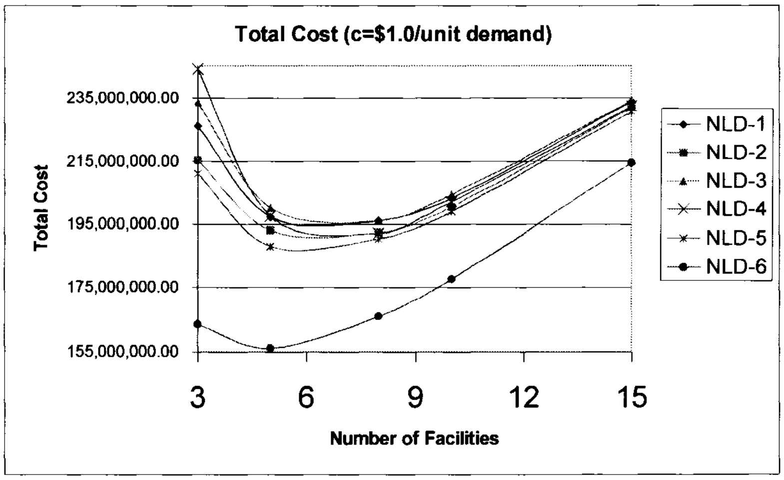

Figure 6.12: Total cost as a function of the number of facilities with transportation cost parameter $c = \$1.00/unit$.

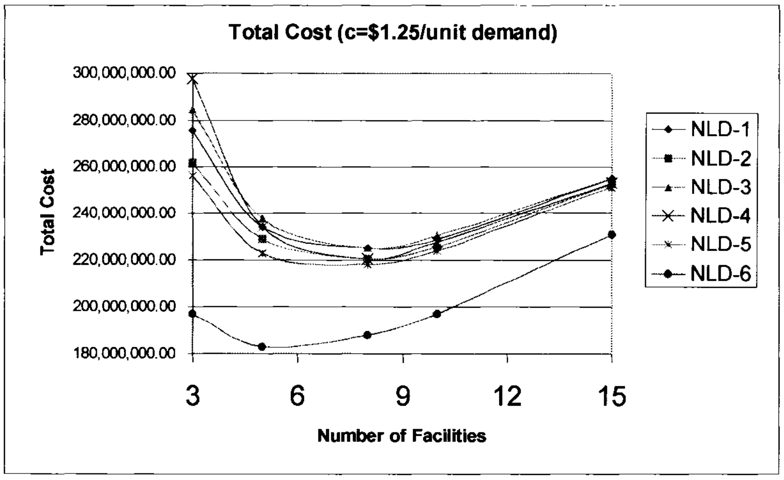

Figure 6.13: Total cost as a function of the number of facilities with transportation cost parameter $c = \$1.25/unit$.

distance than the rapidly varying demand cases.

No. Facilities	NLD-1	NLD-2	NLD-3	NLD-4	NLD-5	NLD-6
3	196,452,765.51	185,407,725.65	203,721,801.60	214,291,715.09	180,896,656.78	133,556,468.51
5	147,242,690.00	143,001,893.93	150,071,764.32	147,318,971.60	137,919,815.99	106,167,105.32
8	115,901,258.78	112,349,753.90	116,056,995.90	112,102,847.79	110,411,142.44	86,112,806.88
10	103,114,048.91	100,360,788.75	104,146,640.91	101,981,543.79	98,967,779.27	77,762,432.44
15	84,025,586.84	82,201,453.71	84,144,840.61	82,583,366.25	80,910,911.55	64,566,055.95

Table 6.8: Total traveled distance results with five different number of facilities when demand is nonlinear

Let's consider Table 6.9. which is calculated from Table 6.8. as before. In comparison with Table 6.7. for linear demand case, Table 6.9. brings about additional findings. The first finding is that there are such cases as $n = 3$ for NLD-3 and NLD-4 where the slowly varying demand would have lower objective than the rapidly varying demand. This special case is attributable to the low number of facilities and the symmetric demand distribution.

Second finding is the extent of the gap between the objective functions of slowly and rapidly varying demand. For example, NDL-5 and NDL-6 has more than 20% difference. Last column in Table 6.9. shows that this difference could well be as high as 60%.[18]. Last additional finding is the impact of the demand density variation on the ideal number of facilities as described in the previous section.

No. Facilities	$\frac{(NLD\text{-}1)-(NLD\text{-}2)}{(NLD\text{-}2)}$	$\frac{(NLD\text{-}3)-(NLD\text{-}4)}{(NLD\text{-}4)}$	$\frac{(NLD\text{-}5)-(NLD\text{-}6)}{(NLD\text{-}6)}$	$\frac{(ROW_{Max}-ROW_{Min})}{ROW_{Min}}$
3	5.62%	-5.19%	26.17%	60.45%
5	2.88%	1.83%	23.02%	41.35%
8	3.06%	3.41%	22.01%	34.77%
10	2.67%	2.08%	21.43%	33.93%
15	2.17%	1.86%	20.20%	30.32%

Table 6.9: Effect of the rate of change in the demand density on the total distance traveled

Let's now turn to the effect of demand density variation on the allocation solutions. Figures 6.14, 6.15 and 6.16 illustrate these results.

Firstly, we discuss Figures 6.14 and 6.15 due to the concave and convex structure of their respective demand densities. From Figure 6.14, we can conclude that higher demand density at the center tend to create more central service regions. This is supported with Figure 6.15 where lower density concentration at the center creates more outside service regions. An interesting observation is the case $n = 8$ in Figure 6.15, where NLD-3 and NLD-4 share similar solution characteristic. This is not the case for concave demand density in Figure 6.14. Also when we rotate NLD-4 figure for $n = 5$ clockwise, we notice the similarity of the solution with NLD-3. In both Figure 6.14 and 6.15, the difference in the size of the service regions is not extreme.

[18]Theoretically, extreme case of the rapidly varying demand is when all demand is concentrated at a single point. In that case, this gap would be infinity.

From Figure 6.16, we further observe that a drastic increase in demand variation results in significant variations of the service region sizes. In addition, high demand variation, with symmetry, tends to favor symmetric allocation regions. Final observation is related to the $n = 8$ cases in NDL-2 and NDL-5. These two allocation solutions are almost identical to each other, i.e. inverted forms.

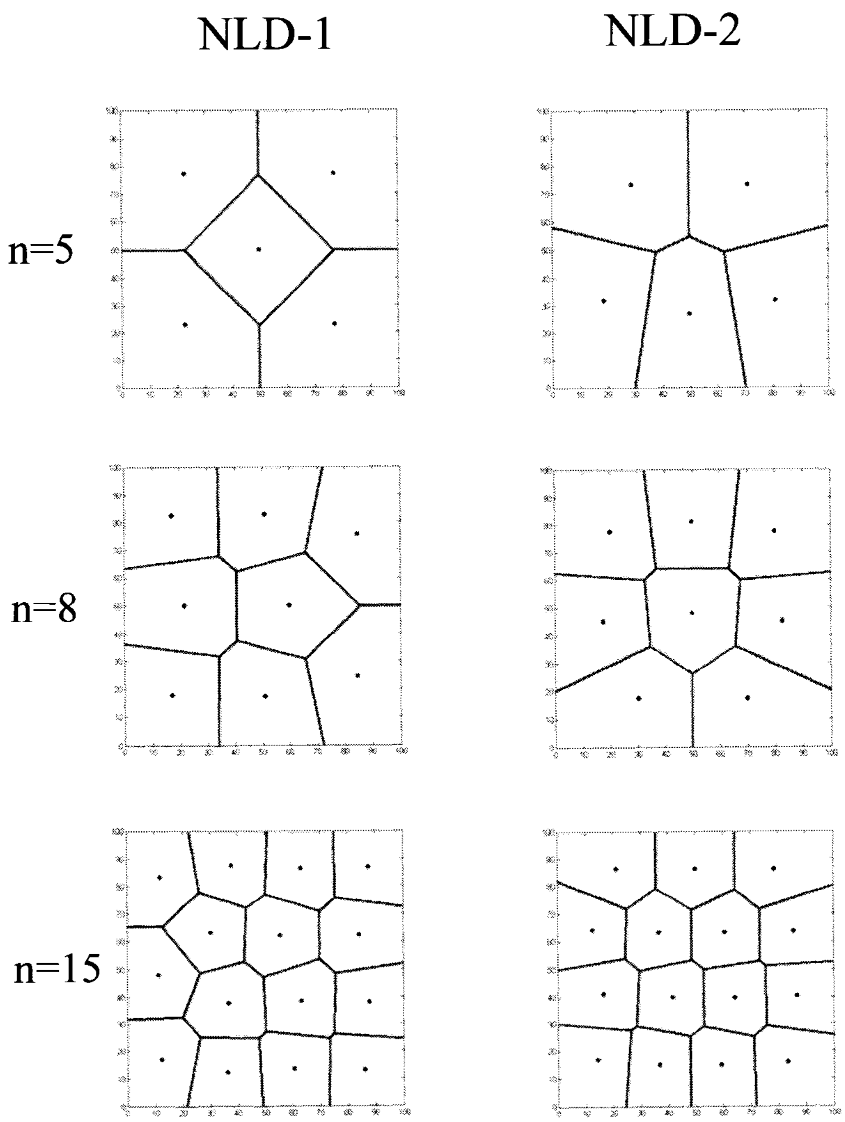

Figure 6.14: Solution to $n = 5, 8,$ and 15 facility cases for demands NLD-1 and NLD-2 (i.e. concave demand).

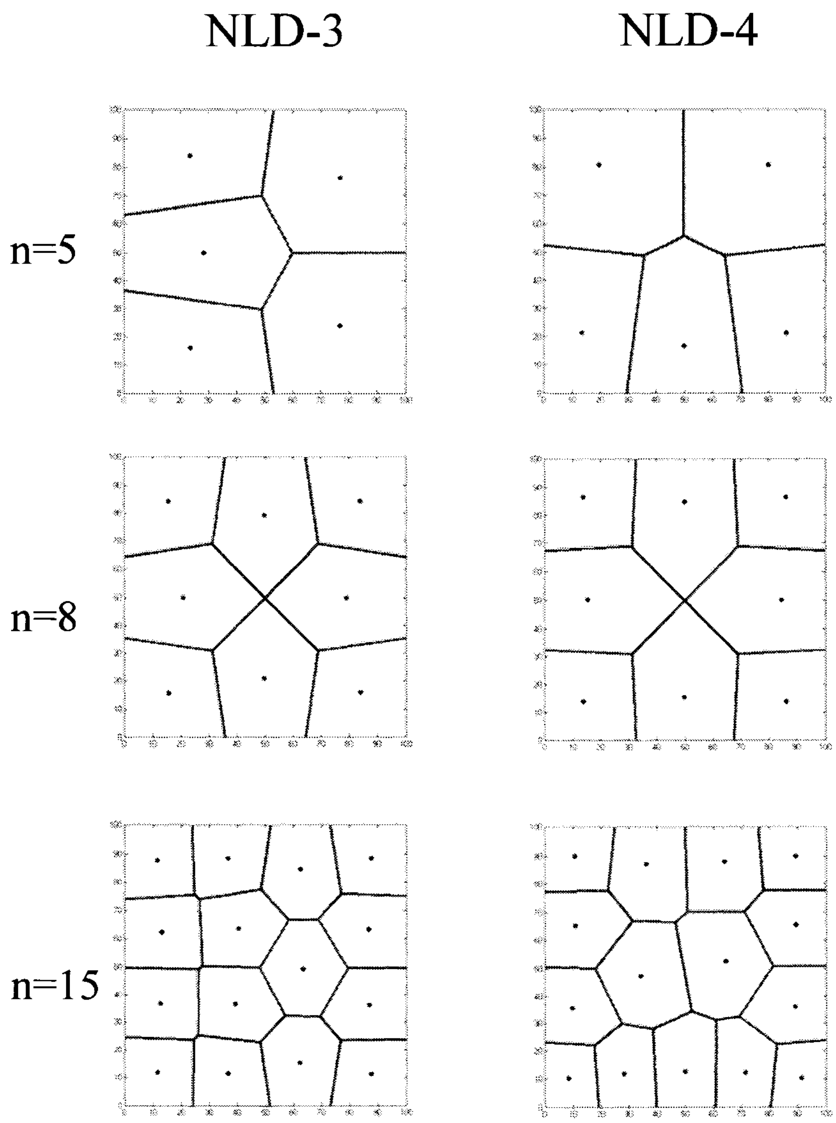

Figure 6.15: Solution to $n = 5, 8$, and 15 facility cases for demands NLD-3 and NLD-4 (i.e. convex demand).

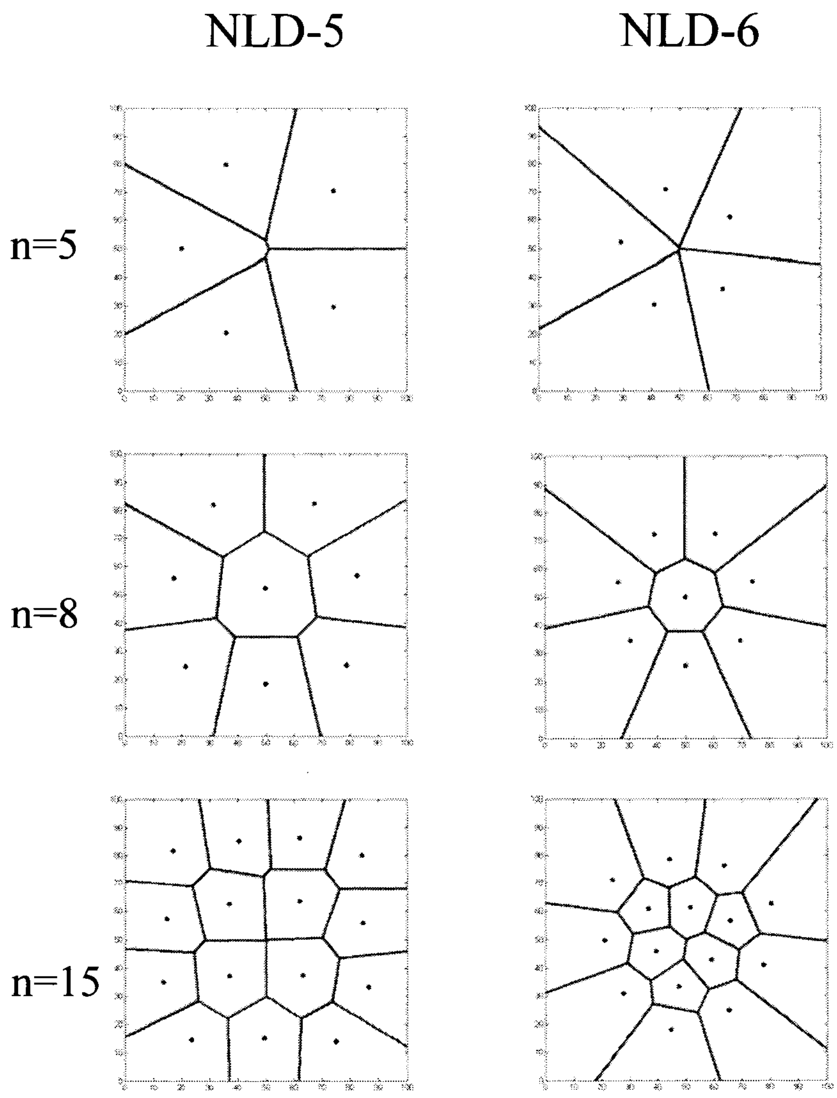

Figure 6.16: Solution to $n = 5, 8,$ and 15 facility cases for demands NLD-5 and NLD-6 (i.e. Newling type demand).

6.6 Conclusions

In this chapter, we have experimented with different demand density functions and number of facilities using vertex-iteration based steepest-descent algorithm developed in Chapter 5. More specifically, we attempt to gain insight on how the rate of change in demand density and the number of facilities affect the objective function as well as the allocation solutions. We conducted this computational study with twelve different demand density functions (six linear and six nonlinear demand density types) and five different number of facilities.

Results from our computational study support our analytical results obtained in Chapter 3 for single dimensional problems. In other words, when the number of facilities is decreased, average size of the service regions increases, which, in turn, **nonlinearly** increases the transportation cost. Furthermore, we have experimented with various transportation cost parameters to understand their impact on the ideal (i.e. lowest total cost) number of facilities. Our results indicate that, for various transportation cost parameters, any reduction from the ideal number of facilities nonlinearly increases the total cost, whereas any increase from this ideal number has an effect of linear increase on the total cost. As we increase the transportation cost parameter, i.e. transportation cost becomes more important relative to fixed costs, then this pattern becomes more pronounced.

In our analysis for the effect of the rate of change in the demand density, we found that slowly varying demand distributions increase the transportation cost. Hence, in a market region where demand is rapidly varying, the transportation costs would be smaller than the case where demand is even across the market region. However, this differential, caused by the rate of change in demand, diminishes with the number of facilities. Our experimental results indicate that this differential could be as high as 60%. Based on this observation as well on the result of previous paragraph, we conclude that, in the absence of accurate demand information, it is better to assume a slowly varying demand. The rationale behind this conclusion is that assuming slowly varying demand induces more facilities than the ideal. Hence the error in total cost by the increased number of facilities is a linear function of the number of facilities (i.e. right side of minimum in Figure 6.7). In contrast, an assumption of rapidly varying demand would induce less number of facilities than the ideal, which would increase the error in total cost nonlinearly (i.e. left side of minimum in Figure 6.7).

Another objective of this chapter is to present results regarding the computational complexity of the vertex-iteration based steepest-descent algorithm. We show that computational complexity of the vertex-iterations is $O(n)$, which is comparable to the efficient voronoi-diagram approach which also has complexity of $O(n)$. Based on our experiment in Section 6.3, number of iterations of the steepest-descent algorithm increases polynomially with the number of facilities. Furthermore, the algorithm requires more iterations with highly nonlinear demand density functions than with linear density functions. Lastly, per iteration runtime increases exponentially with the tolerance parameter of the numerical integration.

Chapter 7

Manhattan Metric: Models and Algorithms

7.1 Introduction

In this chapter, we develop alternative modeling and solution techniques for the location-allocation problems in the allocation variable space based on the Manhattan-metric. This chapter follows the pattern established in the preceding chapters for the Euclidean-metric. Such that, it extends results from the single-dimensional problem setting to planar 2-facility case and finally to n-facility planar problems.

In Section 7.2., we revisit the single dimensional problem and propose two variants for the steepest-descent algorithm based on the iteration of the optimal locations. Since, Euclidean-metric and Manhattan-metric cases are identical for single-dimensional problems, these methods are complementary to the ones presented in Chapter 3. These additional methods lays out the algorithmic framework for planar n-facility case. In section 7.3., we provide models and solution approaches for the planar 2-facility case. Similar to the approach in Chapter 4, we represent planar 2-facility allocation decisions using a construct, which is different than the straight line representation of Chapter 4. We further provide both constructive and improvement based solution approaches using this representation. In Section 7.4., we first discuss why it is not possible to extend either the approaches in Section 7.3. or the vertex-iteration based method in Chapter 5 to the planar Manhattan-metric n-facility problems. Building on the results of Section 7.2., we propose a hybrid-method for solving planar n-facility problems. This hybrid-method is a composite algorithmic mapping of the steepest-descent method and the sequential location-allocation (SLA) approach, which is described in Appendix 4.

7.2 Single-dimension: Alternative Solution Techniques for Manhattan Metric

In single-dimension, Manhattan-metric is equivalent to the Euclidean-metric, thus, all the methods illustrated in Chapter 3 are applicable for the Manhattan metric as well. However, these two metrics differ in the planar n-facility setting. Therefore, when we extend the problem scope to planar n-facility setting, the methods illustrated in Chapter 3 are not as useful for the Manhattan-metric case as they are for the Euclidean-metric case. Accordingly, in this section, we propose two variants of the steepest-descent based solution methods for the single-dimensional problem. As it will be illustrated in the final section of this chapter, the second variant would extend to the planar n-facility case for the Manhattan metric. Hence, the purpose of this section is to lay out the algorithmic framework for planar n-facility cases.

For continuous flow, we summarize the notation for single-dimensional problem with less detail than Chapter 3. As before, we will present algorithms based on the linear demand density function without loss of any generality. It can be easily shown that these algorithms do not rely on the linearity assumption. However, extension to nonlinear demand cases would prevent us from expressing some of the closed form results (e.g. optimal location of a single facility) and would likely result in non-convex problem structure.

Decision Variables

n : number of facilities (service regions)

A_i : area of service region i (i.e. $\mathrm{A}_i = |\mathcal{A}_i|$)

B_i : coordinate of boundary between service regions i and $i-1$

Auxiliary Variables

x_i : coordinate of the facility in service region i

x_i^* : optimal location of single-facility for a given service region $\mathcal{A}_i$

($x_i^* = x_{iM}$ is used to denote median location as the optimal location)

Problem Parameters

$D(x) = u + vx$: demand density at x$\in M$ (items/mile)

F : fixed cost of opening a plant

f : fixed cost component of capacity acquisition cost

a : unit capacity acquisition cost

c : per unit-mile distribution cost

M : size of the single-dimensional market region $\mathcal{M}$

From Chapter 3, recall the location-allocation problem in single dimension in the allocation variable space.

Problem P1

$$\min_{n,A_i,B_i} TC = \sum_{i=1}^{n} TC_i(B_i, A_i) = \sum_{i=1}^{n} \left[\begin{array}{c} F + f + a\mathrm{A}_i D(B_i + \frac{\mathrm{A}_i}{2}) \\ +cK(\frac{v\mathrm{A}_i}{u+vB_i})\mathrm{A}_i^2 D(B_i + \frac{\mathrm{A}_i}{2}) \end{array} \right]$$

$s.t.$

$$B_{i+1} = B_i + \mathrm{A}_i \tag{79}$$

$$B_1 = 0,\ B_{n+1} = M,\ \mathrm{A}_i \geq 0,\ \mathrm{B}_i \geq 0,\ n \geq 0 \textit{ and discrete} \tag{80}$$

where $K(\cdot)$ represents the coefficient for the linear demand density variation.

Due to the additivity and separability of the objective function together with linearity of constraints in problem $P1$, this formulation is amenable for dynamic programming formulation for a given n. Hence we could write the Bellman's equation as in (81) after denoting B_is as the state variables, A_is as the control variables.

$$V(B_i, i) = \underset{B_{i+1}\geq 0, A_i \geq 0, \lambda_i}{minimize} \{TC_i(B_i, A_i) + \lambda_i(B_{i+1} - B_i - A_i) + V(B_{i+1}, i+1)\} \tag{81}$$

$V(B_i, i)$: the cost of optimal allocation decisions starting from B_i and i^{th} facility

A more detailed explanation of this transformation of problem $P1$ to the dynamic programming formulation in (81) and derivation of optimality conditions are provided in Chapter 3. From the results in Chapter 3, the first-order necessary optimality conditions, i.e. Euler equations, are as below.

$$\frac{\partial TC_i(B_i, A_i)}{\partial A_i} + \frac{dTC_{i+1}(B_{i+1}, A_{i+1})}{dA_i} = 0 \tag{82}$$

Using the above notations and first-order optimality condition in (82), we could design steepest-descent improvement algorithm as in Section 3.4.2. In the next section, we present two steepest-descent based improvement algorithms which are different than the one in Chapter 3 in the sense that the iteration variables are **optimal locations** rather than the allocation variables.

Steepest Descent Method for the Optimal Location Decisions

The improvement method described in the Chapter 3 is based on the iteration of allocation decisions, which, starting from an initial solution, are updated according to the optimality condition in (82). We can also start with an initial set of locations and apply a similar procedure. Our rationale behind designing a similar improvement procedure based on the location decisions is to obtain a procedure which could readily be transferred to planar setting (this is discussed more in the next section). Before we describe this location-based improvement solution approach, it is important to differentiate our approach from an earlier work (Iri et al. 1983), which also suggests an improvement based solution methodology based on the locations. We differ from their approach in that we iterate the optimal locations as surrogate iterates of the allocation decisions (i.e. problem in the allocation variable space) whereas theirs is based on location decisions (i.e. problem in the location variable space).

Recall from Chapter 3 that, for a two service-region problem, we were able to identify the optimal allocation solution in a single iteration of steepest descent method (based on the allocation decisions). Whereas the iteration of the optimal-locations as surrogates retains this property, location-space based methods do not. In order to see this, let's consider Figure 7.1 where we have a starting location solution (x_1 and x_2) for a two-area problem. Initial locations of the facilities are at x_1 and x_2, and the border between them is set as B_2 which is at equidistance from these locations. When we evaluate this allocation solution (i.e. B_2) using (82), we observe that the descent direction for x_1 is to the right. The maximum step length for a single improvement iteration of a location-space based method would bring the first facility to the location of x_2, where the second facility is located. It is obvious that this is not the optimal solution as the optimal solution is displayed to the scale in the same figure. In other words, as long as the second facility is located at x_2, the optimal solution is not attainable. However, when we start with optimal locations (x_{1M} and x_{2M}) and iterate x_{1M} as a surrogate of the allocation decision A_1, we would reach to the optimal solution in a single iteration.[19]

In what follows, we describe **two variant of the steepest-descent method** based on the iteration of optimal locations using the information from allocation decisions.

In the **first variant**, given an initial set of locations, we first calculate their implied allocation decisions. Based on these allocations, we identify the optimal locations. Next, we determine the improvement directions for these optimal locations based on the Euler equation in (82) and partial derivative of allocation decisions with respect to the optimal locations. In other words, we are *still optimizing over the allocation variables but performing the line search using the optimal locations implied by these allocations.* Since optimality of these locations with respect to their allocation regions is imposed, allocation decisions are iterated as well.

Now we provide a formal algorithm of the first variant of the steepest-descent method based on optimal location iterations(i.e. median locations) for a given n:

Steepest Descent Solution Algorithm - Independent Optimal Location Iterations (Forward direction)

Step 1. Initialize the model parameters and variables

k : index for optimality iterations

ϵ_{COST} : epsilon parameter for optimality stopping decisions

[19]Note that this is one of the three variants of the steepest descent algrotihm described next. x_{2M} will be located optimally after the iterating x_{1M}.

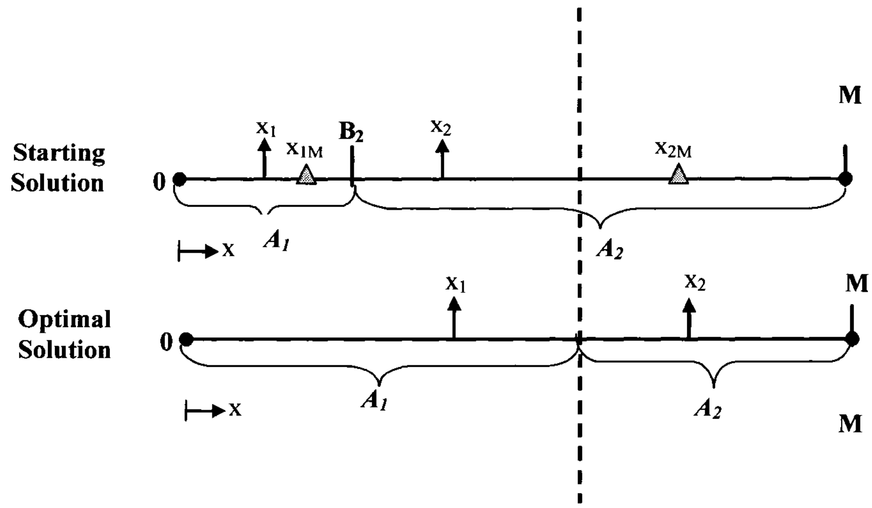

Figure 7.1: Comparison of the improvement based solution approach when the starting solution is given as an initial location decision.

$TC(B_1^0, A_1^0) = \infty$

Define $\vartheta : (x_i^*, B_i) \rightarrow A_i$ as the mapping from a given optimal location and a starting boundary to the service region

$\vartheta : (x_i^*, B_i) \equiv (x_i^*) \rightarrow A_i = \frac{-(u+vB_i)+\sqrt{\left(u+vB_i+x_i^*v\right)^2+\left(x_i^*v\right)^2}}{v}$

Step 2. Start from an initial solution $(B_i^1, A_i^1, i = 1, 2, ..n)$

Do While $(|TC(B_1^k, A_1^k) - TC(B_1^{k-1}, A_1^{k-1})| \geq \epsilon_{COST})$**:**

$k = k + 1$

For $i = 1$ **to** $n - 1$**, Repeat**

• Calculate the optimal location $(x_i^*)^k$by minimizing (6) which is the inverse mapping of ϑ. as follows $(x_i^*)^k = \frac{-2(u+vB_i)+\sqrt{(2u+2vB_i+A_iv)^2+(A_iv)^2}}{2v}$.

• Calculate the steepest direction vector:

$d_{x_i^*}^k = -\left(\frac{\partial TC(B_i, A_i')}{\partial A_i'} + \frac{dTC(B_i+A_i', B_{i+2}-A_i'-B_i)}{dA_i'}\right)\frac{dA_i'}{dx_i^*}$

where $\frac{dA_i'}{dx_i^*} = \frac{2(vB_i+u)+2x_i^*v}{\sqrt{(vB_i+u)^2+4x_i^*v^2B_i+4x_i^*vu+2x_i^{*2}v^2}}$ for $A_i' = A_i^k, B_i = B_i^k, x_i^* = (x_i^*)^k$.

• Solve $\min_{\lambda_i^k} TC(B_i^k, \vartheta(x_i^* + \lambda_i^k d_i^k)) + TC(B_i^k + \vartheta(x_i^* + \lambda_i^k d_i^k), B_{i+2}^k -$

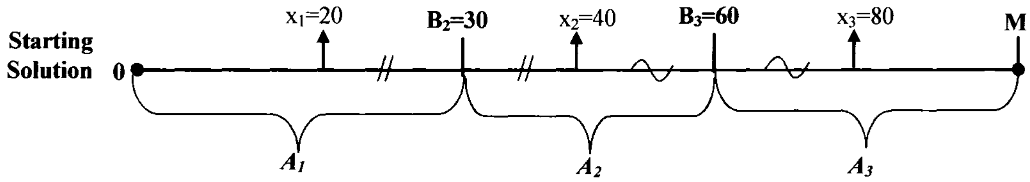

Figure 7.2: Three area example where facilities are initially located at $x_1 = 20,\ x_2 = 40,$ and $x_3 = 80$ (Example 7.1).

$B_i^k - \vartheta(x_i^* + \lambda_i^k d_i^k))$ using a line search method and set $A_i^k = \vartheta(x_i^* + \left(\lambda_i^k\right)^* d_i^k)$.

- Set $B_{i+1}^k = B_i^k + A_i^k$,and $A_{i+1}^k = B_{i+2}^k - A_i^k - B_i^k$.

Return

Step 3. Terminate with the solution $\left(B_i^k, A_i^k, i = 1, 2, ..n\right)$

Example 7.1:

We now illustrate this approach with the example used in Chapter 3. Assume that we are given three initial locations: $x_1 = 20$, $x_2 = 40$, and $x_3 = 80$. Note that these locations correspond to the service regions of size A_1=30, A_2=30, and A_3=40 for the market region defined over $[B_1,M]$ =[0,100] as shown in Figure 7.2.

The demand density function defined over the market is again $D(x) = 10 + 5x$. The median locations are shown with triangles: $x_{1M} = 20.47, x_{2M} = 47.36$, and $x_{3M} = 82.40$. With the facilities at these median locations our problem is locationally optimal but is infeasible in terms of allocation decisions (i.e. $[x_{1M}, B_2] = 9.53 \neq 17.36 = [B_2, x_{2M}]$).

Let's now perform the improvement iterations starting from left boundary ($B_1 = 0$) onwards (i.e. in the forward direction). We first consider $[B_1^1, B_3^1] = [0, 60]$ where $(x_1^*)^1$ =20.67.

Accordingly, $\frac{dA_1'}{dx_1^*}$ =1.4170 and $\frac{\partial TC(B_1,A_1')}{\partial A_1'} = 1492.5$, $\frac{dTC(B_1+A_1',B_3-A_1'-B_1)}{dA_1'} =$

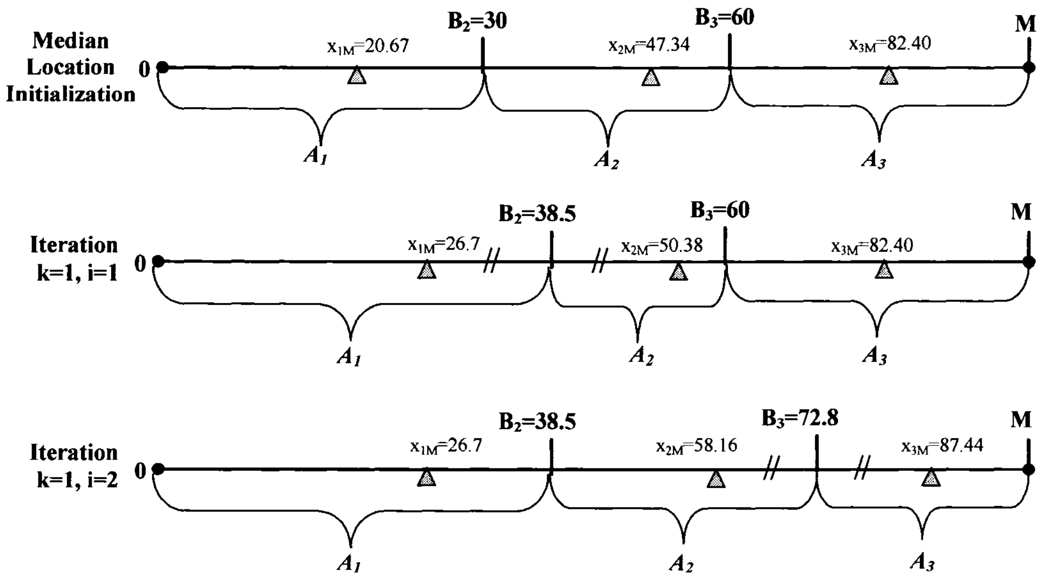

Figure 7.3: First iteration steps for the Example 7.1.

-2773.7 hence d_1^1 =1815.3. When we minimize

$TC(0, \vartheta(20.67 + 1815.3\,(\lambda_1^1))) + TC(\vartheta(20.67 + 1815.3\,(\lambda_1^1)), 60 - \vartheta(20.67 + 1815.3\,(\lambda_1^1)))$, we obtain $\lambda_1^1 = 0.003$. We update $A_1^1 = \vartheta(20.67{+}1815.3\,(0.003))$ =38.54, B_2^1 =38.54, and $A_2^1 = 60 - 38.54 - 0 = 21.46$.

Next we consider $[B_2^1, M] = [38.54, 100]$ where $(x_2^*)^1$ =50.38. Accordingly, $\frac{dA_2'}{dx_2^*}$ =1.6897 and $\frac{\partial TC(B_2,A_2')}{\partial A_2'} = 2981.7$, $\frac{dTC(B_2+A_2',M-A_2'-B_2)}{dA_2'}$ =-6945.2 hence d_2^1 =6697.1. When we minimize $TC(38.54, \vartheta(21.46 + 6697.1\,(\lambda_2^1))) + TC(38.54\vartheta(21.46+6697.1\,(\lambda_2^1)), 100-38.54-\vartheta(21.46+6697.1\,(\lambda_2^1)))$, we obtain λ_2^1 =0.001. We update $A_2^1 = \vartheta(21.46{+}6697.1\,(0.001))$ =34.26, B_3^1 =38.54+34.26=72.80, and $A_3^1 = 100 - 38.54 + 34.26 = 27.20$. This completes the first iteration. $\mathrm{A}_1^1 = 38.54$, A_2^1 =34.26, and $\mathrm{A}_3^1 = 27.20$ and optimal facility locations are the median centers at $x_{1M}^1 = 26.70$, $x_{2M}^1 = 58.16$, and $x_{3M}^1 = 87.44$. Figure 7.3 illustrates the steps of this iteration.

After repeating the same steps in iteration 1, we obtain $\mathrm{A}_1 = 46.90$, $\mathrm{A}_2 =$

28.99, and $A_3 = 24.11$ at the end of iteration 2. Corresponding optimal facility locations are the median centers located at $x^1_{1M} = 32.61$, $x^1_{2M} = 63.03$, and $x^1_{3M} = 88.75$, respectively.

When we repeat, we converge to the optimal solution of $A_1 = 49.60$, $A_2 = 27.35$, and $A_2 = 23.05$ in the sixth iteration. Note that the results of this approach are identical to those in Chapter 3. This is expected, since we are still optimizing the allocation decisions through the optimal facility locations. Complete iteration results are shown in Table 7.1. First column represents the iteration no, i.e. k. Second column is for the service region pairs, i.e. $i = 1$ and $i = 2$ blocks consider A_1 and A_2, and A_2 and A_3, respectively.

(k)	Optimal Location Iteration(i)	A'(i)	B(i)	B(i+1)	B(i+2)	x*(i)	$dTC(A'_i,B_i)/dA'_i$	$dTC(B_{i+2}-A'_i-B_i,B_i+A'_i)/dA'_i$	dA'/dx*(i)	$d_{x^*(i)}$	λ*(i)	$x^*(i)+d_{x^*(i)}\lambda^*(i)$	A(i)
1	1	30.00	0.00	30.00	60.00	20.67	1492.5	-2773.7	1.4170	1815.3	0.003	26.702	38.54
	2	21.46	38.54	60.00	100.00	50.38	2981.7	-6945.2	1.6897	6697.1	0.001	58.162	34.26
2	1	38.54	0.00	38.54	72.80	26.70	2400.0	-3977.1	1.4159	2233.1	0.003	32.605	46.90
	2	25.90	46.90	72.80	100.00	61.19	4342.2	-5475.2	1.6896	1914.3	0.001	63.030	28.99
3	1	46.90	0.00	46.90	75.89	32.61	3494.5	-3944.1	1.4154	636.4	0.002	34.030	48.91
	2	26.98	48.91	75.89	100.00	63.80	4708.6	-5008.1	1.6896	506.1	0.001	64.266	27.76
4	1	48.91	0.00	48.91	76.68	34.03	3789.2	-3908.1	1.4153	168.2	0.002	34.393	49.43
	2	27.25	49.43	76.68	100.00	64.46	4804.3	-4882.3	1.6895	131.8	0.001	64.584	27.45
5	1	49.43	0.00	49.43	76.88	34.39	3866.3	-3897.2	1.4153	43.8	0.002	34.487	49.56
	2	27.32	49.56	76.88	100.00	64.64	4829.2	-4849.4	1.6895	34.2	0.001	64.667	27.37
6	1	49.56	0.00	49.56	76.93	34.49	3886.3	-3894.3	1.4153	11.4	0.002	34.511	49.60
	2	27.34	49.60	76.93	100.00	64.68	4835.7	-4840.9	1.6895	8.9	0.001	64.688	27.35

Table 7.1: Iteration results for improvement based algorithm for independent iteration of optimal-facility locations (Example 7.1).

Second variant differs from the first in that it *iterates all the optimal decisions (x^*_i) at the same time*, as in the case of joint iteration of allocation decisions in Chapter 3. If we assume that each allocation decision ($A_i, i = 1...n-1$) is dependent only on its optimal location as in equation (83), then this approach is indeed equivalent to the joint iteration of the allocation decisions, and it is merely a joint iteration of the optimal locations in the first variant.

$$d_{x_i^*} = d_{A_i} \frac{dA_i}{dx_i^*} \tag{83}$$

where $d_{A_i} = \frac{dTC}{dA_i}$ and $d_{x_i^*} = \frac{\partial TC}{\partial x_i^*}$.

The gradient for x_i^* in (83) implicitly assumes that x_{i+1}^* would be automatically positioned after the determination of A_i, thus x_{i+1}^* is dependent on x_i^* with a known formulae. This assumption is easy to implement in the single dimension, since x_{i+1}^* can be easily expressed in terms of the preceding allocation decision A_i. As we will see in the next section, we cannot follow the same approach in the planar case due to the difficulty in iterating allocation decisions. Instead, we need to consider x_i^*s independently and thus account for the effect of change in A_i on x_{i+1}^*. Therefore, we propose a more proper gradient measure for the optimal location decisions as in (84).

$$d_{x_i^*} = d_{A_{i-1}} \frac{\partial A_{i-1}}{\partial x_i^*} + d_{A_i} \frac{\partial A_i}{\partial x_i^*} \tag{84}$$

At each iteration, we determine a gradient of the total cost with respect to the optimal locations and then perform a line search for the step size. The gradient component for each optimal location ($d_{x_i^*}$) is based on the allocation decision as in (84).

Before presenting the algorithm, we first show the derivation for (84). Consider the allocation decision, A_i, as a function of the optimal locations that it separates (i.e. $\mathrm{A}_i(x_i^*, x_{i+1}^*)$), hence we can express the differential change in the allocation decision as a function of the differential in the optimal locations

x_i^* and x_{i+1}^*.

$$
\begin{aligned}
dA_i &= \frac{\partial A_i}{\partial x_i^*}dx_i^* + \frac{\partial A_i}{\partial x_{i+1}^*}dx_{i+1}^* \\
dTC &= \sum_{i=1}^{n-1} \frac{dTC}{dA_i}dA_i = \sum_{i=1}^{n-1} \frac{dTC}{dA_i}\left(\frac{\partial A_i}{\partial x_i^*}dx_i^* + \frac{\partial A_i}{\partial x_{i+1}^*}dx_{i+1}^*\right) \\
&= \sum_{i=1}^{n-1}\left(\frac{dTC}{dA_i}\frac{\partial A_i}{\partial x_i^*}dx_i^*\right) + \sum_{i=1}^{n-1}\left(\frac{dTC}{dA_i}\frac{\partial A_i}{\partial x_{i+1}^*}dx_{i+1}^*\right) \\
\text{where } TC &= \sum_{i=1..n} TC(B_i, A_i)
\end{aligned}
$$

First relation above assumes that allocation decision between two areas (A_i, A_{i+1}) is **solely** determined by the optimal locations within each area. We can therefore express the partial derivatives of TC with respect to the optimal locations as follows:

$$
\begin{aligned}
\frac{\partial TC}{\partial x_i^*} &= \frac{dTC}{dA_i}\frac{\partial A_i}{\partial x_i^*} + \frac{dTC}{dA_{i-1}}\frac{\partial A_{i-1}}{\partial x_i^*} \qquad i = 2...n-1 \\
\frac{\partial TC}{\partial x_1^*} &= \frac{dTC}{dA_1}\frac{\partial A_1}{\partial x_1^*} \text{ and } \frac{\partial TC}{\partial x_n^*} = \frac{dTC}{dA_{n-1}}\frac{\partial A_{n-1}}{\partial x_n^*}
\end{aligned}
$$

Denoting $d_{A_i} = \frac{dTC}{dA_i}$ and $d_{x_i^*} = \frac{\partial TC}{\partial x_i^*}$, (84) is obtained. For instance, in the two service regions case:

$$
\frac{\partial TC}{\partial x_1^*} = \frac{dTC}{dA_1}\frac{\partial A_1}{\partial x_1^*} \text{ and } \frac{\partial TC}{\partial x_2^*} = \frac{dTC}{dA_1}\frac{\partial A_1}{\partial x_2^*}
$$

When we translate the effect of the change in A_i to x_i^*s and move these optimal locations as surrogate iterates, we have two alternative ways to reallocate. One is to calculate new allocations, i.e. $\mathrm{A}_i + \lambda d_{A_i}$'s, and then relocate facilities optimally given these new allocation decisions. However, in this case we would be again iterating the allocation decisions which, as mentioned above, brings about difficulties in the planar case.

Second alternative is to make an optimal allocation based on the iterated optimal locations, i.e. $\hat{x}_i^*=x_i^*+\lambda d_{x_i^*}$. These optimal allocations are the equidistant allocation decisions, which can be found by calculating $\hat{B}_{i+1} = \frac{\hat{x}_i^*+\hat{x}_{i+1}^*}{2}$ and $\hat{A}_i = \hat{B}_{i+1} - \hat{B}_i$. Note that in this case, there is no guarantee that optimal locations $(\hat{x}_i^*)$ would still be optimal with respect to the allocation decisions $(\hat{A}_i)$. In fact, iterated locations would be optimal with respect to the allocation decisions $(\hat{A}_i)$ if they represent an optimal solution to the problem $P1$. Therefore, in order to return to the original state, where we have optimal locations for a given set of allocation decisions, we perform one last step of optimal locations for the given allocation decisions by $\hat{x}_i^* = \vartheta^{-1}(\hat{A}_i, B_i)$. Here ϑ^{-1} is the mapping to determine optimal location given the allocation decisions $(\hat{A}_i, B_i)$, *i.e.* inverse mapping of ϑ introduced earlier in this section.

It is important to point out that optimal step length (λ^*) is determined based on the final solution where the location decisions are optimal given the allocation decisions. Therefore, in a single iteration step, we move from one solution to another while retaining the optimality of the location decisions. When viewed as a single iteration step, this approach is a single step of a two-step algorithm. We call this approach hybrid approach which would be discussed in more detail in the last section for the planar n-facility case.

We will now provide the algorithm for the joint iteration of optimal locations based on the gradient in (84). The algorithm for this second variant is as follows:

Steepest Descent Solution Algorithm - Joint Optimal Location Iterations (Forward direction)

Step 1. Initialize the model parameters and variables

k : index for the iterations

ϵ_{COST} : epsilon parameter for optimality stopping decisions

$TC(k=0)=\infty$

Step 2. Start from an initial solution $(B_i^1, A_i^1, i=1,2,..n)$

Do While $(|TC(k)-TC(k-1)| \geq \epsilon_{COST})$:

$k=k+1$

- **Gradient Calculation**

  Calculate $d_{x_i^*}^k = d_{A_{i-1}}^k \frac{\partial A_{i-1}^k}{\partial x_i^*} + d_{A_i}^k \frac{\partial A_i^k}{\partial x_i^*}$ for $i=1..n$

  where $x_i^* = (x_i^*)^k$ and,

$$d_{A_i}^k = -\left(\frac{\partial TC(B_i^k, A_i^k)}{\partial A_i^k} + \frac{dTC(B_i^k + A_i^k, B_{i+2}^k - A_i^k - B_i^k)}{dA_i^k}\right)$$

$$\frac{\partial A_i^k}{\partial x_i^*} = \frac{2(u+x_i^* v)}{\sqrt{2(vx_i^*+u)^2-(vB_i^k+u)^2}}$$

$$\frac{\partial A_{i-1}^k}{\partial x_i^*} = \frac{2(u+x_i^* v)}{\sqrt{4(vx_i^*+u)^2-(vA)^2}}$$

- **Parametric tiling**

  $x_i^* = (x_i^*)^k + (\lambda_i^k)\, d_{x_i^*}^k$, $i=1,...,n$.

  $B'_{i+1} = \frac{x_i^* + x_{i+1}^*}{2}$, $i=1,...,n-1$.

  $A'_i = B'_{i+1} - B'_i$, $i=1,...,n$.

- **Line Search for** λ_i^k

  Solve $\min_{\lambda^k} TC = \sum_{i=1}^{n} TC_i(B'_i, A'_i)$ using line search method

- **Finalize tiling**

  $x_i^* = (x_i^*)^k + (\lambda_i^k)^* d_{x_i^*}^k$, $i=1,...,n$.

$$B_{i+1}^{k+1} = \frac{x_i^* + x_{i+1}^*}{2},\ i = 1,\ldots,n-1.$$

$$A_i^{k+1} = B_{i+1}^{k+1} - B_i^{k+1},\ i = 1,\ldots,n.$$

$$(x_i^*)^{k+1} = \vartheta^{-1}(A_i^{k+1})$$

Step 3. Terminate with the solution $\left(B_i^k, A_i^k, i = 1, 2, ..n\right)$

Table 7.2. presents the iteration results for the same example used in the first variant. Compared to the first variant, we have a faster convergence since convergence is attained in fewer iterations. At each iteration step, we perform a line search which involves determining optimal allocation decisions. Even though in the single dimensional case this is a mere arithmetical operation (i.e. $\hat{B}_{i+1} = \frac{\hat{x}_i^* + \hat{x}_{i+1}^*}{2}$), in the planar case this would require a nearest-neighbor search.

Iteration No (k)	x1*(k)	B(2)	x2*(k)	B(3)	x3*(k)	$d_{x1^*(k)}$	$d_{x2^*(k)}$	$d_{x3^*(k)}$	λ*(k)	TC*(k)
0	20.67	30.00	47.34	60.00	82.40	-	-	-	-	228,917.96
1	31.09	47.11	63.14	76.26	89.39	1815.35	6149.79	3107.71	0.0027	189,864.08
2	33.20	48.21	63.21	75.93	88.65	1560.24	-308.00	-899.54	0.0010	187,283.60
3	34.41	49.54	64.68	76.96	89.24	327.43	592.93	216.85	0.0027	186,933.79
4	34.48	49.57	64.66	76.92	89.19	68.35	-38.49	-53.48	0.0008	186,928.32

Table 7.2: Iteration results for improvement based algorithm for joint iteration of optimal-facility locations (Example 7.1).

7.3 Planar Model: 2 Facility Case

In this section, we model the planar 2-facility location-allocation problem in the allocation variable space. We first present the notation, parameters and constructs necessary for representation of the problem. Then, we provide the traditional modeling approach in the location variable space and our alternative model in the allocation variable space. Lastly, we propose two solution approaches, namely **constructive shooting** and **steepest-descent** algorithms, for the location-allocation problem model in allocation variable space.[20]

7.3.1 Description of Parameters and Notation

Although most of the notation and parameters, introduced below, are already presented in Chapter 4 for the Euclidean-metric cases, we repeat them for the continuous flow of the material in this chapter. In the remainder of this section, we will use the following notation and parameter definitions:

Parameters:

$\mathbf{x}$: a point in the two dimensional space $\mathbf{x} \equiv (x, y)$

$\mathcal{M}$: Two dimensional market area (assumed to be a closed and compact set)

$D(\mathbf{x})$: Demand density function over the two-dimensional market region $\mathcal{M}$ $(D(\mathbf{x}) \equiv D(x, y))$

$d_{L_1}(\mathbf{x}_1, \mathbf{x})$: Shortest distance between $\mathbf{x}_1$ and $\mathbf{x}$ based on *Manhattan-metric (L_1)*

In two-dimensional formulations, we have two main decision variables: *Location* decisions and *Allocation* decisions. These decision variables are defined below.

Decision Variables:

$\mathbf{x}_1, \mathbf{x}_2$: locational coordinates of the facilities in service region 1 and 2,

[20] In most part, the notation, parameters and alternative modeling approaches are similar to the Euclidean-metric case of Chapter 4. However, the allocation decision representation and first-order conditions are different for the Manhattan-metric. Reader could comfortably skip the subsections other than these two differences.

i.e. $\mathbf{x}_1 \equiv (x_1, y_1)$, $\mathbf{x}_2 \equiv (x_2, y_2)$)

$\mathcal{A}_1, \mathcal{A}_2$: Service regions 1 and 2 (assumed to be closed sets).

$\mathbf{x}_1^*, \mathbf{x}_2^*$: Optimal locations given the allocation decisions $\mathcal{A}_1$ and $\mathcal{A}_2$

Given the location decisions $(\mathbf{x}_1, \mathbf{x}_2)$, optimal allocation decisions can be found based on the *nearest-neighbor property.* In this case, they would be the point sets defined as follows.

$$\begin{aligned}
\mathcal{A}_1 &= \{\mathbf{x} \mid d_{L_1}(\mathbf{x}_1, \mathbf{x}) \leq d_{L_1}(\mathbf{x}_2, \mathbf{x}),\ \mathbf{x} \in \mathcal{M}\} \\
\mathcal{A}_2 &= \{\mathbf{x} \mid d_{L_1}(\mathbf{x}_2, \mathbf{x}) \leq d_{L_1}(\mathbf{x}_1, \mathbf{x}),\ \mathbf{x} \in \mathcal{M}\}
\end{aligned}$$

When we formulate the location-allocation problem in the allocation variable space, we need to use a more descriptive construct for the allocation decisions than the above point-set definitions. This construct is the **Allocation Line (BR)**, which is defined as follows:

BR : intersection point set of the allocation decisions $\mathcal{A}_1$ and $\mathcal{A}_2$ (i.e. $BR = \mathcal{A}_1 \cap \mathcal{A}_2$). Herein it will be referred as the *Allocation Line.* For the nearest-neighbor solution case, this allocation line could be expressed as below.

$$BR = \{\mathbf{x} \mid d_{L_1}(\mathbf{x}_1, \mathbf{x}) = d_{L_1}(\mathbf{x}_2, \mathbf{x}),\ \mathbf{x} \in \mathcal{M}\}$$

Since we are assuming that service regions are closed sets, all points on the BR must be connected. Whereas the allocation **line** (BR) is a straight line for the Euclidean-metric cases, it has a different form for the Manhattan-metric cases. For the Manhattan-metric distance measure, nearest-neighbor solution of BR consists of at most three straight lines that are parallel to the x-axis,

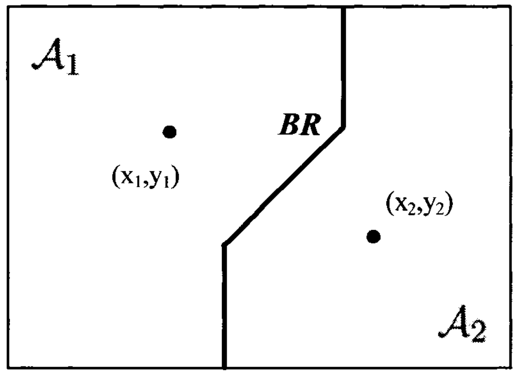

Figure 7.4: Metric dependent alternatives of the separating allocation set BR (a.k.a. Allocation Line).

y-axis, or diagonal lines with angles $\pi/4$ and $3\pi/4$. Figure 7.4 illustrates this form of the allocation line BR. From this point on, we will adopt this form of the allocation line for the Manhattan-metric case.

Depending on the ordering of these facility locations, we would have a diagonal line with either $\pi/4$ or $3\pi/4$ angles, i.e. cases 1 and 2 in Figure 7.5. Furthermore, depending on the absolute differences in the locational facility coordinates, $\mathbf{x}_1$ and $\mathbf{x}_2$, BR would have two horizontal or two vertical segments. This is illustrated in Figure 7.5, i.e. case 1 versus case 3.

In addition to the cases in Figure 7.5, there are two special cases of BR for the L_1 metric. First of these cases is when coordinates of the facilities are identical in either dimension (equivalence in y dimension is shown in Figure 7.6 on the left). Second special case is when the absolute difference in the two facilities' coordinates is equal. In this case, as shown in Figure 7.6 on the right, any point in the shaded corner regions belongs to BR.

The characterization of the BR for the L_1 metric using slope and intercept is more involved than the straight line case. Therefore we distinguish between *diagonal* and *parallel elements* of BR using the **indices d** and **p**, namely BR_d

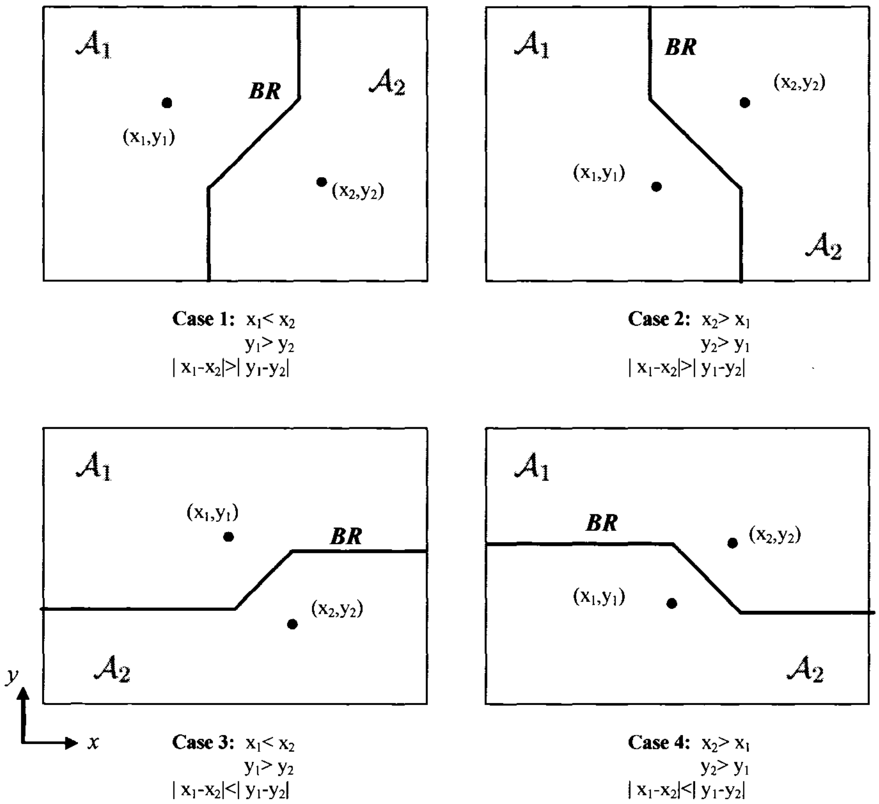

Figure 7.5: Alternatives of the allocation line (BR) for L_1 metric.

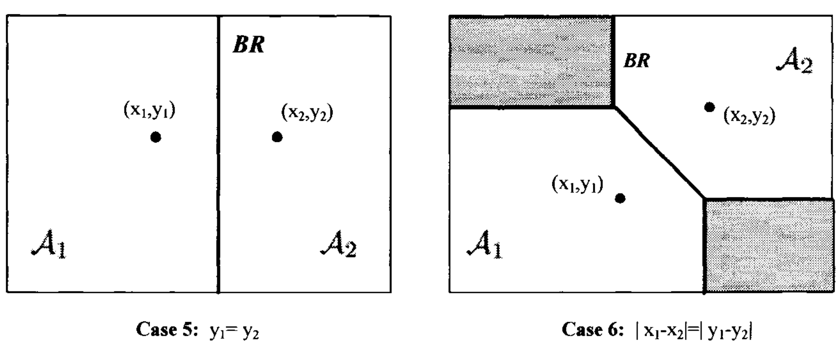

Figure 7.6: Special cases of the allocation line, BR, for L_1 metric.

and BR_p.

$$BR_d = \{\mathbf{x}|\mathbf{x} = (x,y) \in \mathcal{M} \text{ and } y = a_d\, x + b_d\ \}$$

where

$$a_d \;=\; \left\{ \begin{array}{ll} 1 & \text{for } x_1 < x_2 \text{ and } y_1 > y_2 \\ \text{-1} & \text{for } x_1 < x_2 \text{ and } y_1 < y_2 \end{array} \right\}$$

$$b_d \;=\; \left\{ \begin{array}{ll} \frac{y_1 - x_1}{2} + \frac{y_2 - x_2}{2} & \text{for } x_1 < x_2 \text{ and } y_1 > y_2 \\ \frac{y_1 + x_1}{2} + \frac{y_2 + x_2}{2} & \text{for } x_1 < x_2 \text{ and } y_1 < y_2 \end{array} \right\}$$

Case 1: $x_1 < x_2,\ y_1 > y_2$ and $|x_1 - x_2| > |y_1 - y_2|$

$$BR_p = \left\{ \begin{array}{l} \mathbf{x}|\mathbf{x} = (x,y) \in \mathcal{M} \text{ and } x = \frac{x_1 + y_1}{2} + \frac{x_2 - y_2}{2} \text{ for } y > \max(y_1, y_2) \\ \mathbf{x}|\mathbf{x} = (x,y) \in \mathcal{M} \text{ and } x = \frac{x_1 - y_1}{2} + \frac{x_2 + y_2}{2} \text{ for } y < \min(y_1, y_2) \end{array} \right\}$$

Case 2: $x_1 < x_2,\ y_2 > y_1$ and $|x_1 - x_2| > |y_1 - y_2|$

$$BR_p = \left\{ \begin{array}{l} \mathbf{x}|\mathbf{x} = (x,y) \in \mathcal{M} \text{ and } x = \frac{x_1 + y_1}{2} + \frac{x_2 - y_2}{2} \text{ for } y > \max(y_1, y_2) \\ \mathbf{x}|\mathbf{x} = (x,y) \in \mathcal{M} \text{ and } x = \frac{x_1 - y_1}{2} + \frac{x_2 + y_2}{2} \text{ for } y < \min(y_1, y_2) \end{array} \right\}$$

Case 3: $x_1 < x_2,\ y_1 > y_2$ and $|x_1 - x_2| < |y_1 - y_2|$

$$BR_p = \left\{ \begin{array}{l} \mathbf{x}|\mathbf{x} = (x,y) \in \mathcal{M} \text{ and } y = \frac{y_1 - x_1}{2} + \frac{y_2 + x_2}{2} \text{ for } x > \max(x_1, x_2) \\ \mathbf{x}|\mathbf{x} = (x,y) \in \mathcal{M} \text{ and } y = \frac{y_1 + x_1}{2} + \frac{y_2 - x_2}{2} \text{ for } x < \min(x_1, x_2) \end{array} \right\}$$

Case 4: $x_1 < x_2,\ y_2 > y_1$ and $|x_1 - x_2| < |y_1 - y_2|$

$$BR_p = \left\{ \begin{array}{l} \mathbf{x}|\mathbf{x} = (x,y) \in \mathcal{M} \text{ and } y = \frac{y_1+x_1}{2} + \frac{y_2-x_2}{2} \text{for } x > \max(x_1,x_2) \\ \mathbf{x}|\mathbf{x} = (x,y) \in \mathcal{M} \text{ and } y = \frac{y_1-x_1}{2} + \frac{y_2+x_2}{2} \text{ for } x < \min(x_1,x_2) \end{array} \right\}$$

Note that while a closed form expression of $br^{-1}(\cdot)$ exists for BR_d, the same is not true for BR_p. This is because BR_p represents a one-to-many relationship whereas BR_d represents one-to-one relationship.

7.3.2 Alternative Modeling Approaches for 2-Facility Case

In this section we will describe two alternative modeling approaches for planar location-allocation problems; these approaches are also applicable to cases with more than two facilities as we will illustrate in the final section of this chapter. In order to juxtapose these two modeling approaches, namely **Location Variable Space (LVS)** and **Allocation Variable Space (AVS)**, we first present the generic model formulation in the joint variable space. In all of these models, we use $(\mathbf{x}_1,\mathbf{x}_2)$ to denote the location decisions and $A^y(x)$ and $A^x(y)$ to denote the allocation decisions.

The generic model in joint variable space is as follows.

Location-Allocation Model (LAM)[21]:

$$\min_{\substack{A^x(y),A^y(x) \\ \mathbf{x}_1=(x_1,y_1),\mathbf{x}_2=(x_2,y_2)}} \quad TC(A^y(x), A^x(y), \mathbf{x}_1, \mathbf{x}_2)$$

[21] Location-allocation problem, independent of which variable space(s) it is formulated in, is a non-convex problem as illustrated in Chapter 4.

subject to

$$(A^x(y), y) = (x, A^y(x)) \qquad for\ x \in X_{BR},\ y \in Y_{BR}$$

where

$TC(A^y(x), A^x(y), \mathbf{x}_1, \mathbf{x}_2)$: is the total cost function defined over the (2×1) column vectors, $\mathbf{x}_1$ and $\mathbf{x}_2$, and functionals, $A^y(x)$ and $A^x(y)$, defined in the preceding section.

We first present the model in the location variable space, where the allocation decisions are optimized given the location decisions. In the single-dimensional case, optimal allocation decision, which is a single boundary point, could be expressed as $\left(\frac{x_1+x_2}{2}\right)$. However, closed form expressions in two-dimensional setting are difficult to obtain, besides being unnecessary. Instead, we will include this solution in the constraint set. It can be shown that optimal allocation decisions, given locations, would satisfy the following condition.

$$d_{L1}(\mathbf{x}_1, (x, A^y(x))) = d_{L1}(\mathbf{x}_2, (A^x(y), y))$$

for $\forall x \in X_{BR}$, $\forall y \in Y_{BR}$, and $(A^x(y), y) = (x, A^y(x))$.

We now present the *location-allocation problem in the location variable space (LAM-LVS).*

LAM- Location Variable Space (LAM-LVS):

$$\min_{\mathbf{x}_1=(x_1,y_1),\mathbf{x}_2=(x_2,y_2)} TC(A^y(x), A^x(y), \mathbf{x}_1, \mathbf{x}_2)$$

$$s.t.$$

$$\begin{aligned}
(A^x(y), y) &= (x, A^y(x)) \qquad \text{for } x \in X_{BR} \text{ and } y \in Y_{BR} \\
d_{L1}(\mathbf{x}_1, (x, A^y(x))) &= d_{L1}(\mathbf{x}_2, (A^x(y), y)) \qquad \text{for } \forall x \in X_{BR}, \forall y \in Y_{BR}
\end{aligned} \tag{85}$$

The only difference between LAM and LAM-LVS is the last constraint (85). This constraint conditions the optimality of allocation decisions on the location decisions, while making the allocation decisions endogenous decision variables and leaving the location variables as exogenous decision variables. The following proposition establishes the first-order necessary condition for the LAM-LVS given the allocation decisions (A_i).

Proposition 7.1

The optimal locations of the two facilities ($\mathbf{x}_1^*$ and $\mathbf{x}_2^*$), *given the allocation decisions, satisfy the following conditions when the distance measure is based on the* $\mathbf{Manhattan - Metric\ (L_1)}$:

$$\int_y \int_{x<x_i^*,\ x\in\mathcal{A}_i} D(\mathbf{x})d\mathbf{x} = \int_y \int_{x\geq x_i^*,\ x\in\mathcal{A}_i} D(\mathbf{x})d\mathbf{x} \qquad \text{for } i = 1, 2$$

$$\int_x \int_{y<y_i^*,\ y\in\mathcal{A}_i} D(\mathbf{x})d\mathbf{x} = \int_x \int_{y\geq y_i^*,\ y\in\mathcal{A}_i} D(\mathbf{x})d\mathbf{x} \qquad \text{for } i = 1, 2$$

Proof.

Proof can be found in Appendix 7. ■

7.3.3 Modeling in Allocation Variable Space

Next we present the model in the allocation variable space, where the location decisions are optimized given the allocation decisions.

LAM- Allocation Variable Space (LAM-AVS):

$$\min_{A^x(y),A^y(x)} TC(A^y(x), A^x(y), \mathbf{x}_1^*, \mathbf{x}_2^*)$$

$$s.t.$$

$$\begin{aligned} (A^x(y), y) &= (x, A^y(x)) \qquad \text{for } x \in X_{BR} \text{ and } y \in Y_{BR} \\ \mathbf{x}_i^* &= \arg\min_{(\mathbf{x}_i)} \int_{\mathcal{A}_i} d_{L_1}(\mathbf{x}_i, \mathbf{x}) D(\mathbf{x}) d\mathbf{x} \qquad \text{for } i = 1, 2 \end{aligned} \tag{86}$$

Optimal solution to (86) can be expressed in closed form for the single-dimensional case, but same is not true for the two-dimensional case. In particular, these optimal location solutions satisfy the first order necessary condition of the LAM-LVS outlined in the Proposition 7.1. When these necessary conditions are included in the constraint set of the LAM, we obtain the above *location-allocation model in the allocation variable space* (LAM-AVS). First order necessary condition for LAM-AVS (assuming the presence of only the transportation costs in the objective) is as below.

First Order Necessary Conditions for LAM-AVS

$$d_{L1}(\mathbf{x}_1^*, (x, A^y(x))) - d_{L1}(\mathbf{x}_2^*, (A^x(y), y)) = 0 \tag{87}$$

for $\forall x \in X_{BR}$, $\forall y \in Y_{BR}$, and $(A^x(y), y) = (x, A^y(x))$

7.3.4 Solution Methodologies for 2-Facility Case

Constructive- Shooting Algorithm

Constructive solution approach for the Manhattan metric is similar to the Euclidean-metric based methods except the number of equations to be solved. Since Euclidean-metric based allocation line is a straight line, which can be characterized by its slope and intercept, the constructive solution approach for Euclidean metric requires solution of only two equations. In comparison, Manhattan-metric based allocation line is made up of three special components which can be fully characterized by three parameters. A general form of this type of allocation line is illustrated in Figure 7.7. The two special cases in the figure, horizontal and vertical parallel components, are determined by the absolute differences of the x- and y-dimensional location coordinates as explained before. Without loss of generality, herein we will adapt the allocation line on the left in Figure 7.7.[22]

First, we will derive the differential equations based on the first-order conditions of the LAM-AVS for Manhattan-metric. Next, a formal presentation of the shooting algorithm based on the vertical parallel components (case on the left in Figure 7.7) will be presented. Lastly, this section will conclude with an example application of the shooting algorithm for the Manhattan metric.

As illustrated in Chapter 3, we can express the objective function in either

[22] Recall from the case of Euclidean-metric that when our shooting levels are not accurate, shooting algorithm would not converge. In the case of Manhattan-metric, if we assume the two parallel components of BR as vertical lines whereas they are indeed horizontal at the optimal, then shooting algorithm would not converge. Then we would change two-components from vertical to horizontal lines, which is similar to the change over from horizontal shooting to vertical shooting in the Euclidean-metric case.

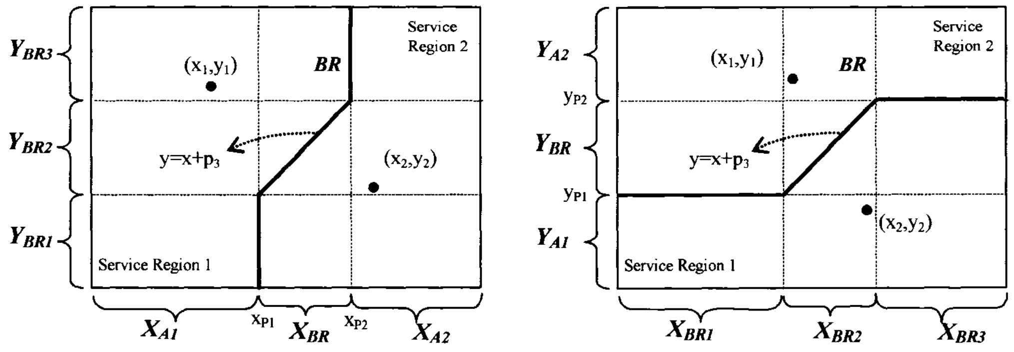

Figure 7.7: Allocation line form when distance measure is based on Manhattan-metric

horizontal single-dimensional decisions ($A^x(y)$) or vertical single-dimensional decisions ($A^y(x)$). For the Manhattan-metric, we choose the horizontal single-dimensional allocation decision, $A^x(y)$, as the unknown variables. As mentioned before, Manhattan-metric case requires three differential equations for three $A^x(y)$ unknowns. Hence we arbitrarily choose three $y - axis$ values (y_{P1}, y_{P2}, y_{P3}) to define the variables, namely $A^x(y_{P1}), A^x(y_{P2})$, and $A^x(y_{P3})$. These variables must satisfy the following equations, which are derived from the first-order conditions of the LAM-AVS in (87).[23]

$$|x_1^* - A^x(y_{Pj})| + |y_1^* - y_{Pj}| - |x_2^* - A^x(y_{Pj})| - |y_1^* - y_{Pj}| = 0 \qquad for\ j = 1, 2, 3 \tag{88}$$

For ease of exposition, let's define the following notation as illustrated in Figure

[23]These first order conditions are as a result of choosing $A^x(y)$ as the decision variable and using the following first-order condition $\frac{dTC}{dA^x(y)} = \frac{\partial TC}{\partial A^x(y)} + \frac{\partial TC}{\partial A^y(x)}\frac{\partial A^y(x)}{\partial A^x(y)} = 0$. For a more detailed explanation, readers could refer to Chapter 3.

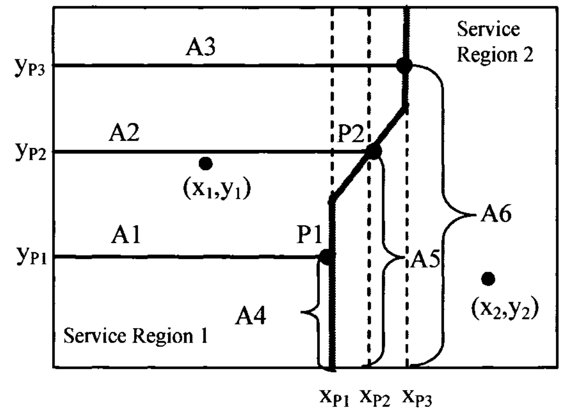

Figure 7.8: Initial triggers for the constructive solution approach (Manhattan-metric based measure)

7.8.

$$A1 \; : \; = A^x(y_{P1}) \text{ and } A2 := A^x(y_{P2}) \text{ and } A3 := A^x(y_{P3})$$
$$A4 \; : \; = A^y(x_{P1}) \text{ and } A5 := A^y(x_{P2}) \text{ and } A6 := A^y(x_{P3})$$

Shooting algorithm philosophy for the Manhattan-metric is same as in the Euclidean-metric based case, thus for brevity we do not repeat here. For algorithmic efficiency, Newton-Raphson method of updating is used for all three triggers. One difference is that with the Manhattan metric, allocation decisions $\mathcal{A}_1$ and $\mathcal{A}_2$ are fully characterized when the initial triggers $(A1, A2, A3)$ are decided. So we do not need to account for special cases in the same capacity as in the Euclidean-metric based shooting algorithm. Lastly, since LAM-AVS problem is in the allocation variable space, (x_1^*, y_1^*) and (x_2^*, y_2^*) satisfy the following conditions which are repeated for convenience.

$$\int_y \int_{x<x_i^*, x\in\mathcal{A}_i} D(\mathbf{x})d\mathbf{x} = \int_y \int_{x\geq x_i^*, x\in\mathcal{A}_i} D(\mathbf{x})d\mathbf{x} \quad \text{for } i=1,2 \tag{89}$$

$$\int_x \int_{y<y_i^*, y\in\mathcal{A}_i} D(\mathbf{x})d\mathbf{x} = \int_x \int_{y\geq y_i^*, y\in\mathcal{A}_i} D(\mathbf{x})d\mathbf{x} \quad \text{for } i=1,2 \tag{90}$$

Shooting Algorithm: Manhattan-metric Based Distance Measure (L_1)

Step 1. Define and Initialize the model parameters and variables

j : index for the feasibility iterations

(i.e. $j^* = \{j|\ \epsilon_{BOUND} \geq |M1' - M|,\ \epsilon_{BOUND} \geq |M2' - M|$, and $\epsilon_{BOUND} \geq |M3' - M|\}$

ϵ_{BOUND} : epsilon parameter for feasibility stopping decision

h : centered difference approximation parameter for partial differentials

$A1^j$: j^{th} iteration estimate for the first service region size at y_{P1}

$A2^j$: j^{th} iteration estimate for the first service region size at y_{P2}

$A3^j$: j^{th} iteration estimate for the first service region size at y_{P3}

$\mathbf{x}_i^* = (x_i^*, y_i^*)$: optimal locations corresponding to $\mathcal{A}_{i=1,2}^j$

$(M')^j$: boundary variable (i.e. $M^j = M$ is the feasible boundary condition)

$M1', M2', M3'$: solutions for M^j in (88) for $y_{Pj=1,2,3}$

Set $j=0$, $A1^{j=1}$, $A2^{j=1}$ and $A3^{j=1}$

Step 2. Update the first service region sizes $A1^j$,$A2^j$ and $A3^{j=1}$

Do While ($|M1'-M| \geq \epsilon_{BOUND}$ and $|M2'-M| \geq \epsilon_{BOUND}$ and $\epsilon_{BOUND} \geq$

$|M3' - M|$):

$j = j + 1$

Step 2.1. Calculate $p_3 = A5 - A2$ and define piecewise functional form of allocation line BR

$$br(y) = \left\{ \begin{array}{cc} A1^j & y \leq A1^j + p_3 \\ y - p_3 & A1^j + p_3 \leq y \leq A3^j + p_3 \\ A3^j & y \geq A3^j + p_3 \end{array} \right\}$$

Step 2.2. Parametrize $\mathcal{A}_1$ and $\mathcal{A}_2$ in terms of $(M')^j$

$Y_{BR1} = \{y | y \in [0, A1^j + p_3]$

$Y_{BR2} = \{y | y \in [A1^j + p_3, A2^j + p_3]$

$Y_{BR3} = \{y | y \in [A2^j + p_3, M]$

$\mathcal{A}_1 := \{(x, y) | y \in Y_{BR1}$ and $x \in [0, A1^j] \cup y \in Y_{BR2}$ and $x \in [0, y - p_3] \cup y \in Y_{BR3}$ and $x \in [0, A2^j]\}$

$\mathcal{A}_2 := \{(x, y) | y \in Y_{BR1}$ and $x \in [A1^j, (M')^j] \cup y \in Y_{BR2}$ and $x \in [y - p_3, (M')^j] \cup y \in Y_{BR3}$ and $x \in [A2^j, (M')^j]\}$

Step 2.3. Solve the following single facility location problems in terms of $(M')^j$

$$\mathbf{x}_1^* = (x_1^*, y_1^*) := \arg \min_{\mathbf{x}_1 = (x_1, y_1)} \left(\int_{\mathcal{A}_1} |\mathbf{x}_1 - \mathbf{x}| \; D(\mathbf{x}) d\mathbf{x} \right)$$

$$x_2^* := \arg \min_{x_2} \left(\int_{\mathcal{A}_2} |x_2 - x| \; D(\mathbf{x}) d\mathbf{x} \right)$$

Step 2.4. Assign y_2^* equidistant value based on x_1^*, y_1^*, and x_2^* and solve following for boundary value $M1'$

$$y_2^* := \left\{ \begin{array}{ll} |x_1^* - A1| + |y_1^* - A4| - |x_2^* - A1| + A4 & y_2^* \geq A4 \\ -|x_1^* - A1| - |y_1^* - A4| + |x_2^* - A1| + A4 & y_2^* \leq A4 \end{array} \right\}$$

$$\frac{1}{2}\int_{\mathcal{A}_2} D(\mathbf{x})d\mathbf{x} = \int_{y=0}^{y=\frac{y_2^*}{2}} \int_{x=br(y)}^{M1'} D(\mathbf{x})d\mathbf{x}$$

Step 2.5. Assign y_2^* equidistant value based on x_1^*, y_1^*, and x_2^* and solve following for boundary value $M2'$

$$y_2^* := \left\{ \begin{array}{ll} |x_1^* - A2| + |y_1^* - A5| - |x_2^* - A2| + A5 & y_2^* \geq A5 \\ -|x_1^* - A2| - |y_1^* - A5| + |x_2^* - A2| + A5 & y_2^* \leq A5 \end{array} \right\}$$

$$\frac{1}{2}\int_{\mathcal{A}_2} D(\mathbf{x})d\mathbf{x} = \int_{y=0}^{y=\frac{y_2^*}{2}} \int_{x=br(y)}^{M2'} D(\mathbf{x})d\mathbf{x}$$

Step 2.6. Assign y_2^* equidistant value based on x_1^*, y_1^*, and x_2^* and solve following for boundary value $M3'$

$$y_2^* := \left\{ \begin{array}{ll} |x_1^* - A3| + |y_1^* - A6| - |x_2^* - A3| + A6 & y_2^* \geq A6 \\ -|x_1^* - A3| - |y_1^* - A6| + |x_2^* - A3| + A6 & y_2^* \leq A6 \end{array} \right\}$$

$$\frac{1}{2}\int_{\mathcal{A}_2} D(\mathbf{x})d\mathbf{x} = \int_{y=0}^{y=\frac{y_2^*}{2}} \int_{x=br(y)}^{M3'} D(\mathbf{x})d\mathbf{x}$$

Step 2.7. Calculate $\mathcal{F}_1$ and $\mathcal{F}_2$

$\mathcal{F}_1(A1^j, A2^j, A3^j) = M1' - M$

$\mathcal{F}_2(A1^j, A2^j, A3^j) = M2' - M$

$\mathcal{F}_3(A1^j, A2^j, A3^j) = M3' - M$

Step 2.8. Approximate J using centered difference approximation

Repeat Steps 2.1-2.6 for $(A1^j - h, A2^j, A3^j)$,$(A1^j + h, A2^j, A3^j)$, $(A1^j, A2^j - h, A3^j)$, $(A1^j, A2^j + h, A3^j)$, $(A1^j, A2^j, A3^j - h)$, $(A1^j, A2^j, A3^j + h)$

Step 2.9. Assign

$$\begin{pmatrix} A1 \\ A2 \\ A3 \end{pmatrix}^{j+1} = \begin{pmatrix} A1 \\ A2 \\ A3 \end{pmatrix}^{j} - J^{-1} \begin{pmatrix} F_1(A1^j, A2^j, A3^j) \\ F_2(A1^j, A2^j, A3^j) \\ F_3(A1^j, A2^j, A3^j) \end{pmatrix}$$

where $J = \begin{bmatrix} \frac{\partial F_1}{\partial A1} & \frac{\partial F_1}{\partial A2} & \frac{\partial F_1}{\partial A3} \\ \frac{\partial F_2}{\partial A1} & \frac{\partial F_2}{\partial A2} & \frac{\partial F_2}{\partial A3} \\ \frac{\partial F_3}{\partial A1} & \frac{\partial F_3}{\partial A2} & \frac{\partial F_3}{\partial A3} \end{bmatrix}$

Return.

Step 3. Terminate with the solution $A1^j$, $A2^j, A3^j$ and BR

Note that in **Step 2.3**, (x_1^*, y_1^*) are identified numerically and x_2^* is parametrized over M'. Reason for leaving y_2^* to **Steps 2.4 to 2.6** is because of the dependency of the optimality condition on $A4, A5$ and $A6$. If we had chosen the case on right in Figure 7.7, then similar procedure would have been applied for x_2^*. Next, we provide an illustrative example for the application of the shooting algorithm for Manhattan-metric.

Example 7.2: Shooting Algorithm- Manhattan-metric Based Distance Measure (L_1)

ITERATION 1

Step 1. Define and Initialize the model parameters and variables

Set: $j = 1, h = 0.1, M = \{(x, y)|, x \in [0, 100] \text{ and } y \in [0, 100]\}$, $\epsilon_{BOUND} = 0.01$

$y_{P1} = A4 = 30$, $y_{P2} = A5 = 50$ and $y_{P3} = A6 = 80$

$A1^{j=1}$=30, $A2^{j=1}$=40 and $A3^{j=1}$ =50

Step 2. Update the first service region sizes $A1^j$,$A2^j$ and $A3^j$

Do While ($|M1' - M| \geq \epsilon_{BOUND}$ and $|M2' - M| \geq \epsilon_{BOUND}$ and $\epsilon_{BOUND} \geq |M3' - M|$):

$j = j + 1$

Step 2.1.

$p_3 = A5 - A2^j = 50 - 40 = 10$

$$br(y) = \left\{ \begin{array}{ll} 30 & y \leq 40 \\ y - 10 & 40 \leq y \leq 60 \\ 50 & y \geq 60 \end{array} \right\}$$

Step 2.2. Parametrize $\mathcal{A}_1$ and $\mathcal{A}_2$ in terms of $(M')^j$

$Y_{BR1} = \{y|y \in [0, 40]$

$Y_{BR2} = \{y|y \in [40, 60]$

$Y_{BR3} = \{y|y \in [60, 100]$

$\mathcal{A}_1 := \{(x, y)|y \in Y_{BR1} \text{ and } x \in [0, 30] \ \cup \ y \in Y_{BR2} \text{ and } x \in [0, y - 10] \cup \ y \in Y_{BR3} \text{ and } x \in [0, 50]\}$

$\mathcal{A}_2 := \{(x, y)|y \in Y_{BR1} \text{ and } x \in [30, (M')^j] \ \cup \ y \in Y_{BR2} \text{ and } x \in [y - 10, (M')^j] \cup \ y \in Y_{BR3} \text{ and } x \in [50, (M')^j]\}$

Step 2.3. Solve the following single facility location problems in terms

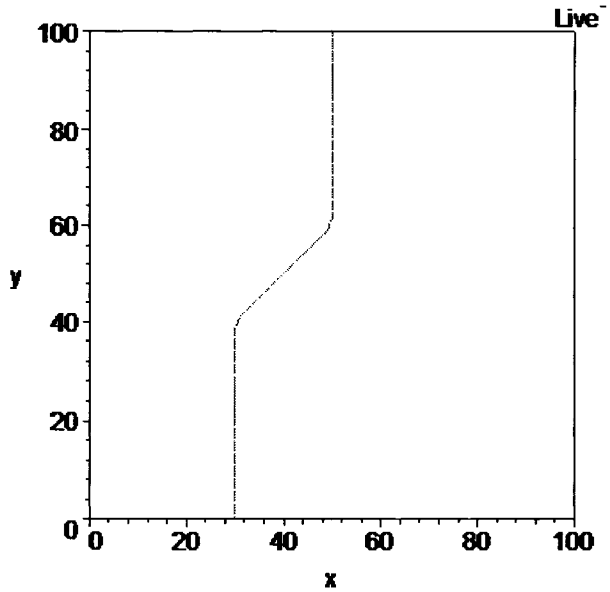

Figure 7.9: Allocation decision in the beginning of first iteration of the shooting algorithm for Manhattan-metric (Example 7.2).

of $(M')^j$

$$
\begin{aligned}
(x_1^*, y_1^*) \quad &: = \arg \min_{\mathbf{x}_1=(x_1,y_1)} \left(\int_{\mathcal{A}_1} |\mathbf{x}_1 - \mathbf{x}| \; D(\mathbf{x}) d\mathbf{x} \right) \\
&= (24.9305, 69.4035)
\end{aligned}
$$

$$
\begin{aligned}
x_2^* \quad &: = \arg \min_{x_2} \left(\int_{\mathcal{A}_2} |x_2 - x| \; D(\mathbf{x}) d\mathbf{x} \right) \\
&= -35 + \frac{(\sqrt{129300 + 1260M' + 18(M')^2})}{6}
\end{aligned}
$$

Step 2.4. $M1'$ =72.76897530

Step 2.5. $M2'$ =98.45515141

Step 2.6. $M3'$ =77.89286118

Step 2.7. Calculate F_1 and F_2

$$F_1(A1^{j=1}, A2^{j=1}, A3^{j=1}) = M1' - 100 = \text{-}27.231$$

$$F_2(A1^{j=1}, A2^{j=1}, A3^{j=1}) = M2' - 100 = \text{-}1.545$$

$$F_3(A1^{j=1}, A2^{j=1}, A3^{j=1}) = M3' - 100 = \text{-}22.107$$

Step 2.8. Approximate J using centered difference approximation

$$J = \begin{bmatrix} \frac{\partial F_1}{\partial A1} & \frac{\partial F_1}{\partial A2} & \frac{\partial F_1}{\partial A3} \\ \frac{\partial F_2}{\partial A1} & \frac{\partial F_2}{\partial A2} & \frac{\partial F_2}{\partial A3} \\ \frac{\partial F_3}{\partial A1} & \frac{\partial F_3}{\partial A2} & \frac{\partial F_3}{\partial A3} \end{bmatrix} = \begin{bmatrix} 1.242 & \text{-}0.332 & 0.437 \\ \text{-}1.007 & 2.266 & 0.194 \\ \text{-}0.868 & \text{-}1.487 & 3.757 \end{bmatrix}$$

Step 2.9. Assign

$$\begin{pmatrix} A1 \\ A2 \\ A3 \end{pmatrix}^{j=2} = \begin{pmatrix} 30.0 \\ 40.0 \\ 50.0 \end{pmatrix}^{j=1} - J^{-1} \begin{pmatrix} -27.231 \\ -1.545 \\ -22.107 \end{pmatrix} = \begin{pmatrix} 49.326 \\ 48.112 \\ 63.558 \end{pmatrix}$$

When we repeat for seven iterations, the algorithm converges to the solution illustrated in Figure 7.10. Results of these iterations are displayed in Table 7.3.

Iteration(j)	A1	A2	A3	A4	A5	A6	P_3	M1'	M2'	M3'	F_1	F_2	F_3
1	30.000	40.000	50.000	30	50	80	10.000	72.77	98.46	77.89	27.231	1.545	22.107
2	49.326	48.112	63.558	30	50	80	1.888	100.53	85.46	91.68	-0.532	14.539	8.319
3	47.701	53.557	70.518	30	50	80	-3.557	99.91	111.20	102.78	0.093	-11.202	-2.776
4	47.884	53.811	70.282	30	50	80	-3.811	99.33	115.27	101.11	0.669	-15.273	-1.108
5	47.203	47.153	68.292	30	50	80	2.847	99.25	90.87	99.02	0.746	9.135	0.983
6	48.140	49.851	69.228	30	50	80	0.149	100.31	100.63	100.58	-0.308	-0.627	-0.583
7	47.910	49.513	68.926	30	50	80	0.487	100.00	100.00	100.00	0.000	0.000	0.001

Table 7.3: Constructive solution algorithm results for initial triggers $A_1 = 30$, $A_2 = 40$ and $A_3 = 50$ with L_1 (Example 7.2)

Improvement Based- Steepest-Descent Algorithm

Next, we will present the steepest-descent improvement algorithm for the

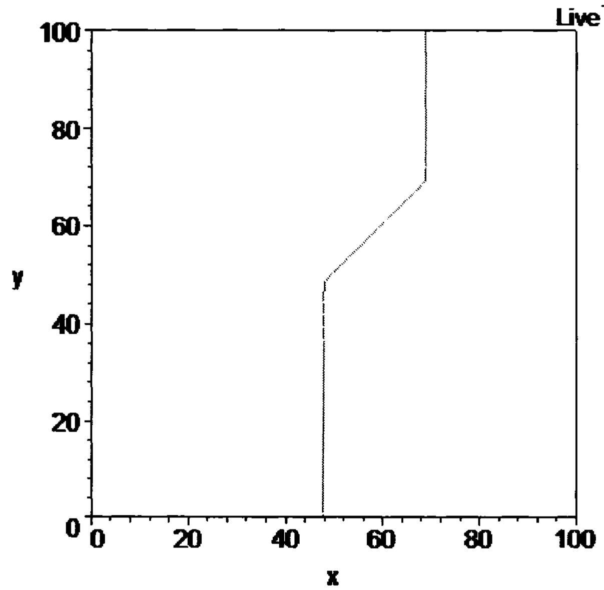

Figure 7.10: Allocation decision at the end of seventh iteration of the shooting algorithm for Manhattan-metric (Example 7.2).

Manhattan-metric case. For this, we first outline the details of the algorithm and use Figure 7.11 as the reference for algorithmic description and its example implementation.

In the case of Manhattan-metric, rotation type of transformation is not shape preserving. In a more formal definition, Manhattan-metric based allocation line is not shape invariant with respect to rotation. However, it is shape invariant with respect to translation, but translation of BR as a singleton would not access all feasible allocation solutions for the Manhattan-metric. Instead, one could treat each of the three parts of BR independently and define a shape preserving transformation for each of these three parts. For instance, in Figure 7.12 , the two vertical segments could translate sidewise and the diagonal could translate vertically. Similarly, for the case on the right in Figure 7.7, horizontal segments could translate vertically and diagonal element could translate horizontally. With such transformation pattern, the aforemen-

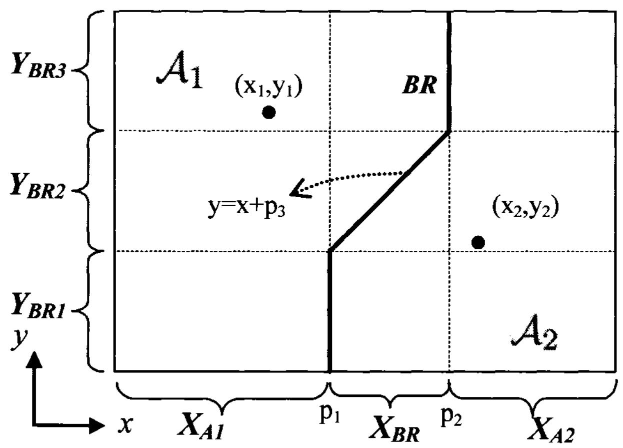

Figure 7.11: Representative solution and illustration of allocation line BR parameters for the Manhattan-metric.

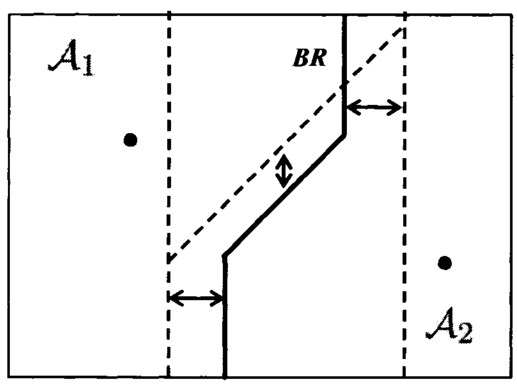

Figure 7.12: Shape preserving transformation for Manhattan-metric-based allocation decisions.

tioned shape for the Manhattan-metric's allocation line BR shape is preserved (see Figure 7.12). Our improvement algorithm is thus developed according to independent iteration of these three components. There are many ways to parametrize the allocation line BR, hence many alternatives of these three components. We, herein, choose the parameters $p1$, $p2$ and $p3$ as illustrated in Figure 7.11.

In comparison with the notation used for constructive approach for Manhattan-metric in Figure 7.8, following relations exist with the improvement based approach. Note that we no longer require the definition of $A2$ used in Figure 7.8, but rather will use the $p3$, the intercept of the diagonal component of BR.

$$\begin{aligned} p1 &\equiv A1 \\ p2 &\equiv A3 \end{aligned}$$

Steepest-Descent Improvement Algorithm: Case $p = L_1$

Step 1. Define and Initialize the model parameters and variables

j : index for optimality iterations (i.e. $j^* = \{j |\ \epsilon_{COST} \geq \frac{|TC^{j+1} - TC^j|}{|TC^j|}\}$

ϵ_{COST} : epsilon parameter for optimality stopping decision

α^j : step length for line search iterations at the jth iteration

p_i^j : j^{th} iteration value for the i^{th} component

$\mathbf{x}_i^* = (x_i^*, y_i^*)$: optimal locations corresponding to $\mathcal{A}_{i=1,2}^j$

M : market boundary parameter

Set $j = 0$, $p_1^{j=0}, p_2^{j=0}$ and $p_3^{j=0}$

Step 2. Initialization: Allocate the service regions and optimally locate facilities

Step 2.1. Define piecewise functional form of allocation line BR

$$br(y) = \left\{ \begin{array}{cc} p_1^j & y \leq p_1^j + p_3^j \\ y - p_3^j & p_1^j + p_3^j \leq y \leq p_2^j + p_3^j \\ p_2^j & y \geq p_2^j + p_3^j \end{array} \right\}$$

Step 2.2. Identify the following sets and functions given BR

$Y_{BR1} = \{y | y \in [0, p_1^j + p_3^j]$

$Y_{BR2} = \{y | y \in [p_1^j + p_3^j, p_2^j + p_3^j]$

$Y_{BR3} = \{y | y \in [p_2^j + p_3^j, M]$

$\mathcal{A}_1 := \{(x, y) | y \in Y_{BR1}$ and $x \in [0, p_1^j] \ \cup \ y \in Y_{BR2}$ and $x \in [0, y - p_3^j] \cup \ y \in Y_{BR3}$ and $x \in [0, p_2^j]\}$

$\mathcal{A}_2 := \{(x, y) | y \in Y_{BR1}$ and $x \in [p_1^j, M] \ \cup \ y \in Y_{BR2}$ and $x \in [y - p_3^j, M] \cup \ y \in Y_{BR3}$ and $x \in [p_2^j, M]\}$

Step 2.3. Find the optimal locations and calculate the total cost

$$\mathbf{x}_1^* = (x_1^*, y_1^*)^j := \arg \min_{(x_1, y_1)} (\int_{\mathcal{A}_1} d_{p=L_1}(\mathbf{x}_1, \mathbf{x}) D(\mathbf{x}) d\mathbf{x})$$
$$\mathbf{x}_2^* = (x_2^*, y_2^*)^j := \arg \min_{(x_2, y_2)} (\int_{\mathcal{A}_2} d_{p=L_1}(\mathbf{x}_2, \mathbf{x}) D(\mathbf{x}) d\mathbf{x})$$
$$TC^j = \int_{\mathcal{A}_1} d_{p=L_1}(\mathbf{x}_1^*, \mathbf{x}) D(\mathbf{x}) d\mathbf{x} + \int_{\mathcal{A}_2} d_{p=L_1}(\mathbf{x}_2^*, \mathbf{x}) D(\mathbf{x}) d\mathbf{x}$$

Step 3. Improvement: Update the first service region sizes $A1^j$ and $A2^j$

Do While $(\frac{|TC^{j+1} - TC^j|}{|TC^j|} \geq \epsilon_{COST})$:

$j = j + 1$

Step 3.1. Calculate the partial gradients

· Total cost with respect to p_1, p_2 and p_3:

$\frac{dTC}{dp_1}, \frac{dTC}{dp_2}, \frac{dTC}{dp_3}$ found in **Appendix 7**

· Normalize the gradients

$$d_{p_i}^j = \frac{\frac{dTC}{dp_i}}{\sqrt{\sum_{i=1,2,3} \frac{dTC}{dp_i}}}$$

Step 3.2. Perform a line search for step size α^j

· Update the two allocation decisions $A1^j$ and $A2^j$

$(p_i^j)' = p_i^j + \alpha^j d_{p_i}^j \qquad$ for $i = 1, 2, 3$

· Repeat **Steps 2.1, 2.2, 2.3** using $(p_i^j)'$

· Find $(\alpha^j)^* = \arg \min_{\alpha^j} TC$

Step 3.3. Update the allocation decisions

$p_i^{j+1} = p_i^j + (\alpha^j)^* d_{p_i}^j \qquad$ for $i = 1, 2, 3$

· Repeat **Steps 2.2, 2.3** using p_i^{j+1} for $i = 1, 2, 3$

· Return to **Step 3**

Step 4. Terminate with the solution p_i^j and BR

Example 7.3: Steepest-Descent Improvement Algorithm - Manhattan-metric (L_1) Case

Let's consider an example implementation of the above algorithm for the Figure 7.11. We will use the same example in the preceding section, namely constructive solution approach. We have a square-shaped market region $\mathcal{M}$= $\{(x,y)|x \in (0,100)$ and $y \in (0,100)\}$, i.e. $M = 100$. We wish to determine an optimal allocation decision for a linear demand density function ($D(x,y) = 100 + 10x + 5y$) over the market region $\mathcal{M}$. The starting solution for this instance is $A1$=35 and $A2 = 40$ at $y_{P1} = A3 = 40$ and $y_{P2} = A4 = 50$, respectively.

ITERATION 1

Step 1. Define and Initialize the model parameters and variables

Set: $j = 0, h = 0.1, M = \{(x,y)|, x \in [0,100]$ and $y \in [0,100]\}, \epsilon_{COST}$: 1×10^{-5}

$$
\begin{aligned}
p_1^{j=0} &= 30 \\
p_2^{j=0} &= 50 \\
p_3^{j=0} &= 10
\end{aligned}
$$

Step 2. Initialization: Allocate the service regions and optimally

locate facilities

Step 2.1. Define piecewise functional form of allocation line BR

$$br(y) = \begin{Bmatrix} 30 & y \leq 40 \\ y - 10 & 40 \leq y \leq 60 \\ 50 & y \geq 60 \end{Bmatrix}$$

Step 2.2. Identify the following sets and functions given BR

$Y_{BR1} = \{y | y \in [0, 40]\}$

$Y_{BR2} = \{y | y \in [40, 60]\}$

$Y_{BR3} = \{y | y \in [60, 100]\}$

$\mathcal{A}_1 := \{(x, y) | y \in Y_{BR1}$ and $x \in [0, 30] \;\cup\; y \in Y_{BR2}$ and $x \in [0, y - 10] \cup\; y \in Y_{BR3}$ and $x \in [0, 50]\}$

$\mathcal{A}_2 := \{(x, y) | y \in Y_{BR1}$ and $x \in [30, 100] \;\cup\; y \in Y_{BR2}$ and $x \in [y - 10, 100] \cup\; y \in Y_{BR3}$ and $x \in [50, 100]\}$

Step 2.3. Find the optimal locations and calculate the total cost:

$\mathbf{x}_1^* = (x_1^*, y_1^*)^{j=0} := (24.93, 82.06)$

$\mathbf{x}_2^* = (x_2^*, y_2^*)^{j=0} := (74.96, 50.60)$

$TC^{j=0} = 323, 252, 518.60$

Step 3. Improvement: Update the first service region sizes $A1^j$ **and** $A2^j$

$j = 1$

Step 3.1. Calculate the partial gradients

· Total cost with respect to p_1, p_2 and p_3:

$$\begin{aligned}\frac{dTC}{dp_1} &= -168,522.5\\ \frac{dTC}{dp_2} &= -798,366.4\\ \frac{dTC}{dp_3} &= -118,182.8\end{aligned}$$

· Normalize the gradients

$$\begin{aligned}d_{p_1}^{j=1} &= -0.20440\\ d_{p_2}^{j=1} &= -0.96834\\ d_{p_3}^{j=1} &= -0.14334\end{aligned}$$

Step 3.2. Perform a line search for step size α^j

$(\alpha^j)^* = \arg\min_{\alpha^j} TC$ =16.74

Step 3.3. Update the allocation decisions :

$$\begin{pmatrix} p_1^{j=1} \\ p_2^{j=1} \\ p_3^{j=1} \end{pmatrix} = \begin{pmatrix} 30 \\ 50 \\ 10 \end{pmatrix} - (16.74) \begin{pmatrix} -0.20440 \\ -0.96834 \\ -0.14334 \end{pmatrix} = \begin{pmatrix} 33.42 \\ 66.21 \\ 12.40 \end{pmatrix}$$

Allocation decisions at the end of iteration 1 is displayed in Figure 7.13. Again, for brevity, we do not detail the remainder of iterations. Table 7.4. presents the results for the remaining iterations. Last column, **% Change**, is the percentage improvement in the current iteration over the incumbent solution. Optimality condition (less than 0.001% improvement) is reached at $j = 10^{th}$ iteration, which is illustrated in Figure 7.14.

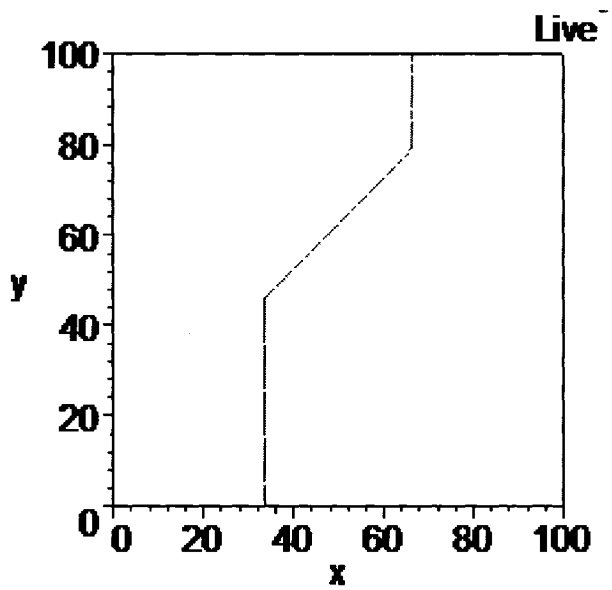

Figure 7.13: Solution at the end of first iteration of the Manhattan-metric based example for Steepest-descent improvement method (Example 7.3).

j	P_1	P_2	P_3	x1	y1	x2	y2	Step Size	TC	% Change
0	30.00	50.00	10.00	24.93	82.06	74.96	50.60	16.74	323,252,518.60	-
1	33.42	66.21	12.40	29.50	74.24	77.52	47.23	8.43	310,736,238.00	4.0279%
2	39.75	67.00	6.89	32.12	71.62	79.04	47.25	5.11	307,776,686.90	0.9616%
3	43.84	67.95	3.97	33.77	70.16	80.02	47.59	2.55	306,743,866.60	0.3367%
4	45.82	68.23	2.39	34.56	69.35	80.49	47.86	1.51	306,461,768.70	0.0920%
5	46.96	68.28	1.40	35.00	68.84	80.76	47.99	0.91	306,366,908.00	0.0310%
6	47.59	68.30	0.74	35.25	68.52	80.91	48.01	0.55	306,333,426.00	0.0109%
7	47.91	68.35	0.30	35.40	68.36	81.00	47.96	0.33	306,320,362.00	0.0043%
8	48.04	68.43	0.01	35.49	68.29	81.06	47.86	0.19	306,314,324.40	0.0020%
9	48.07	68.52	-0.16	35.55	68.29	81.09	47.77	0.13	306,311,300.70	0.0010%
10	48.06	68.60	-0.26	35.58	68.30	81.11	47.69	-	306,309,564.20	0.0006%

Table 7.4: Steepest-descent improvement algorithm's iteration results Euclidean-metric (L_1) example (Example 7.3).

7.4 Planar Model: n Facility Case

7.4.1 Hybrid Approach: Optimal location-based Improvement

In the planar n-facility case for Euclidean-metric case, Chapter 5, we presented two solution methods: Steepest-descent and conjugate-gradient meth-

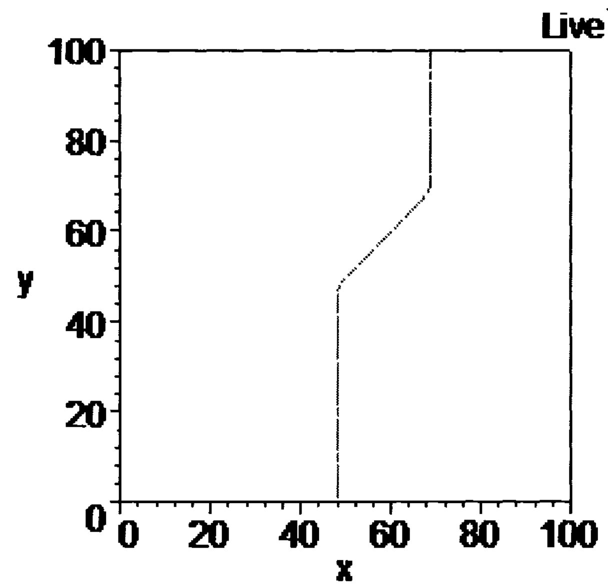

Figure 7.14: Solution at the end of 10^{th} iteration of the Manhattan-metric based example for Steepest-descent improvement method (Example 7.3).

ods. Both of these methods require service regions to be separated by straight lines (i.e. allocation lines or voronoi edges) which is a property satisfied by the Euclidean-metric based measures. Hence, the straight line form of the allocation decisions allowed us to parametrize these lines with their slope and intercept parameters and develop vertex iteration based improvement methods.[24] However for the cases where the service regions are not separated by lines but rather with more complex forms, it is nearly impossible to parametrize the allocation decisions. One example is the Manhattan-metric case where the separation of the service regions are in a special structure as described in the previous section. Figure 7.15 illustrates a typical allocation solution based on the nearest-neighbor property for 5-facility case for the Manhattan metric. Since the relationship between the vertices and the allocation lines cannot be

[24]For a detailed explanation of the voronoi-diagram concepts and vertex-iteration based approach, we refer the reader to Chapter 5.

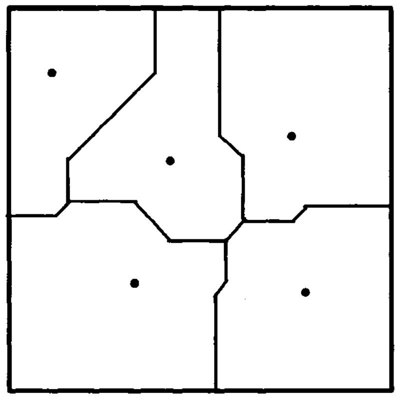

Figure 7.15: Illustration of the allocation solution for 5-facility case based on the Manhattan-metric.

characterized as easily as in the case of Euclidean-metric, we cannot follow the vertex iteration based approaches introduced in the Chapter 5. Instead we will follow a different approach which solves the LAM-AVS (location-allocation problem in the allocation variable space) using information from the incumbent allocation decisions. This approach has been introduced at the beginning of this chapter for the single dimensional problem. The second variant of the steepest-descent method, which combines steepest-descent with one-step optimal location and one-step optimal allocation, is the method we extend to the planar n-facility case.

Our approach is based on the composition of the allocation improvement approach and the sequential location-allocation (SLA) method which is presented in the appendix of Chapter 4.[25] The SLA method is a form of cyclic-coordinate method where the method sequentially solves the problem in one variable space, either in the location variable space or in the allocation variables space, while fixing the other variable set. In that respect, an improvement based approach to the LAM-AVS is a different class of algorithm than the

[25]Reader unfamiliar with the Sequetial Location-Allocation method for location-allocation problems is referred to Appendix 4 for a brief treatment of the method.

SLA method. An improvement based approach applied to LAM-AVS moves from an allocation solution to another, while the locations are kept **optimal**. Therefore, given a solution where allocation decisions are not optimal but locations are optimal, we could proceed with either an improvement approach using the LAM-AVS or perform, first, the optimal allocation step, then the optimal location step of the SLA method. These two alternatives would both lead to solutions where locations are optimal, but allocation decisions are not necessarily optimal.

Now consider the case where we alternate between these two alternatives: given an optimal location solution, we first perform an improvement iteration, i.e. steepest descent, on the allocation variables, and in the second stage we perform, first, the optimal allocation and then the optimal location step of the SLA. As a result, we would again have a solution where the locations are optimal but allocations are not necessarily optimal. Since first step is an improving step as is the second step, the composite approach would improve the objective function.

This sequence of alternating between an improvement method applied to LAM-AVS and then the SLA implementation is at the core of our hybrid approach. Since we are using the SLA method's steps in reverse order, first the optimal allocation and then optimal location step, we could refer to this implementation more appropriately as the sequential allocation-location (SAL) method. Note that so far we have presented these two steps of the hybrid approach as a sequential implementation of the allocation improvement and SAL approaches, meaning that each step is independently performed from each other.

Our **hybrid improvement algorithm** is the composition of these two steps and moves from one solution to the next one after performing first the allocation improvement step and then the SAL. In order to illustrate this, Figure 7.16 would be instructive. This figure represents the contour plot of the objective function in the two-dimensional representation of the location and allocation variable spaces (i.e. $x - axis$ is the allocation variables and $y - axis$ is the location variables). At the top of figure, the path followed by the steepest-descent improvement approach is illustrated. Note that this path corresponds to two steepest descent iterations. In the middle of Figure 7.16, one step implementation of the SAL approach, i.e. optimal allocation and then optimal location, is illustrated. At the bottom, we tie the SAL and steepest-descent approach in a sequence, i.e. one step of SAL followed by one step of steepest-descent, thus this sub-figure also represents two iterations.

In comparison with the **sequential** SAL and steepest-descent approach (bottom of Figure 7.16), Figure 7.17 displays the **composition** of these two approaches. As shown in Figure 7.17, at every iteration, we are moving in the steepest-descent direction and performing one step of SAL. The dashed arrow from the starting point to the final point illustrates this single iteration. Given an allocation decision with optimal locations, we first identify the steepest-descent direction for the allocation decisions. Next, we check whether moving in that direction leads to a better solution than the direct implementation of the SAL. This direct implementation is illustrated in the figure with the dashed arrows representing the SAL steps when we start from the current solution. As we iterate in the steepest-descent direction and repeat the same SAL steps, we reach to an improved solution (second set of dashed arrows representing SAL). Therefore, steepest descent direction in combination with the SAL step dominates the solutions reached by direct implementation of the

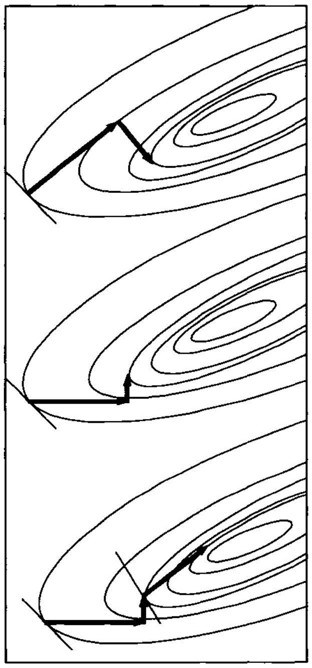

Figure 7.16: Illustration of the allocation improvement approach (top figure), the SLA method (middle figure) and the sequential implementation of these two methods (bottom figure) in the allocation-location variable space. ($x-axis$ is the allocation variables and $y-axis$ is the location variables)

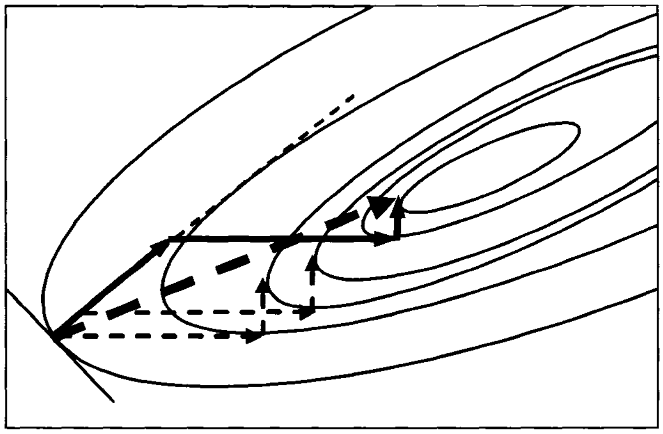

Figure 7.17: Illustration of the Hybrid algorithm as the composition of SAL and steepest-descent approach.

SAL approach. With this information, we perform a line search along the steepest-descent direction using this composite mapping. In another example, it can be shown that steepest-descent direction may lead to an inferior solution compared to the solution obtained by direct implementation of SAL. In this case, a zero step length is the best solution, i.e. direct implementation of the SAL approach.[26]

If we look at this algorithmic mapping as a composition of two mappings, then the follow-up question would be whether this is a closed mapping and, therefore, can we establish its convergence? The answer is yes, and for more information on the composition of mappings and the global convergence theorem we refer the reader to (Luenberger 1984).

Another question is how we will perform the first step, i.e. steepest-descent for allocations, of this composite algorithm (i.e. hybrid algorithm), since im-

[26]In our algorithm, we further control this decision between performing a hybrid-algorithm step and a pure SAL by a desired improvement factor. When the hybrid algorithm with a small step size does not exceed the pure SAL step by a certain factor, we skip the hybrid-step and perform a pure SAL step. This is reflects the tradeoff between the cost of hybrid-algorithm's line-search iterations and potential improvement over pure SAL step.

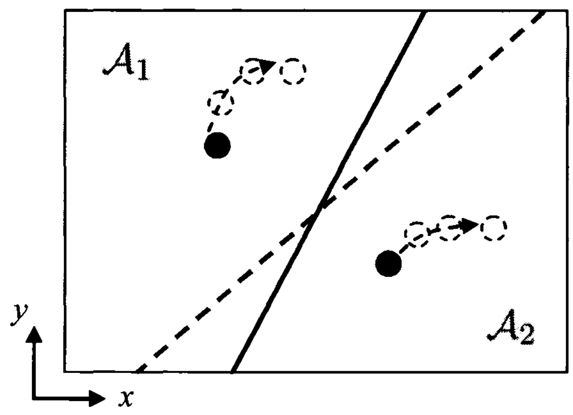

Figure 7.18: Illustration of the relationship between the allocation decisions and optimal locations.

provement approaches for the LAM-AVS require allocation decisions to be in particular structure, namely service regions are separated by lines. The answer lies in the relationship between the allocation decisions and their implied optimal locations. In other words, instead of iterating the allocation decisions, we will iterate the optimal locations as surrogate iterates as done in the second variant of Section 7.2. Figure 7.18 is illustrative for understanding this relationship between the iteration of allocation decisions and the corresponding optimal locations.

In this figure, solid line and the filled circles represents ex-ante solution, and the dashed line and circles represent improved solution. The dashed arrows show the paths followed by the optimal locations as the allocation decision is iterated using an improvement solution approach applied to LAM-AVS. It is possible to identify these paths with Taylor expansion of the relationship between the allocation solution and its corresponding optimal locations. First order Taylor expansion is the method we have chosen here, since second order expansion brings about additional computational complexity.

In what follows, we first establish the linkage between the allocation decisions and their corresponding optimal locations. Then we present the formal hybrid-improvement algorithm and illustrate its implementation with an example.

Optimal Locations

In the first step of our hybrid improvement algorithm, we improve the allocation decisions by iterating the optimal locations. As stated earlier, the path followed by the optimal locations could be approximated with Taylor series and that we herein choose to use first order approximation. First order approximation of the change in optimal locations in terms of changes in the allocation decisions is obtained from the optimality conditions presented in the Proposition 7.1. The following proposition, establishes this relationship for the Manhattan-metric.

Proposition 7.2

The differential change in the optimal locations with respect to single dimensional allocation decisions satisfy the following conditions when the distance measure is based on the $\mathbf{Manhattan-Metric}\ (\mathbf{L}_1)$:

$$
\begin{aligned}
\frac{\partial x_i^*}{\partial A_i^x(y)} &= \frac{D(B_i^x(y)+A_i^x(y),y)}{2D(x_i^*,y)} \qquad \text{for } i=1,2,...,n \\
\frac{\partial y_i^*}{\partial A_i^y(x)} &= \frac{D(x,B_i^y(x)+A_i^y(x))}{2D(x,y_i^*)} \qquad \text{for } i=1,2,...,n
\end{aligned}
$$

$$\frac{\partial x_i^*}{\partial A_{i-1}^x(y)} = \frac{D(B_i^x(y), y)}{2D(x_i^*, y)} \qquad \text{for } i = 1, 2, ..., n$$
$$\frac{\partial y_i^*}{\partial A_{i-1}^y(x)} = \frac{D(x, B_i^y(x))}{2D(x, y_i^*)} \qquad \text{for } i = 1, 2, ..., n$$

Proof.

Proof is provided in the Appendix 7 ■

As illustrated in Chapter 4, we can express the objective function in either horizontal single-dimensional decisions ($A^x(y)$) or vertical single-dimensional decisions ($A^y(x)$). By using the representation based on the horizontal single-dimensional decisions ($A^x(y)$), $\frac{\partial TC}{\partial x_i^*}$ can be expressed as follows:

$$\frac{\partial TC}{\partial x_i^*} = \int_{y \in Y_{BR}} \left(\frac{\partial TC}{\partial A_{i-1}^x(y)} \frac{\partial A_{i-1}^x(y)}{\partial x_i^*} + \frac{\partial TC}{\partial A_i^x(y)} \frac{\partial A_i^x(y)}{\partial x_i^*} \right) dy \qquad (91)$$

Similarly, with the vertical single-dimensional decisions ($A^y(x)$) representation, $\frac{\partial TC}{\partial y_i^*}$ can be expressed as follows:

$$\frac{\partial TC}{\partial y_i^*} = \int_{x \in Y_{BR}} \left(\frac{\partial TC}{\partial A_{i-1}^x(y)} \frac{\partial A_{i-1}^x(y)}{\partial y_i^*} + \frac{\partial TC}{\partial A_i^x(y)} \frac{\partial A_i^x(y)}{\partial y_i^*} \right) dx \qquad (92)$$

where

$$\frac{\partial TC}{\partial A_i^x(y)} = \begin{bmatrix} (|x_i^* - (B_i^x(y) + A_i^x(y))| + |y_i^* - y|) - \\ \left(\left|x_j^* - (B_i^x(y) + A_i^x(y))\right| - \left|y_j^* - y\right|\right) \end{bmatrix} D\left(A_i^x(y), y\right) \qquad (93)$$

$$\frac{\partial TC}{\partial A_i^y(x)} = \begin{bmatrix} (|x_i^* - x| + |y_i^* - (B_i^x(y) + A_i^x(y))|) - \\ \left(\left|x_j^* - x\right| - \left|y_j^* - (B_i^x(y) + A_i^x(y))\right|\right) \end{bmatrix} D\left(x, A_i^y(x)\right) \qquad (94)$$

Note that (91) and (92) are planar versions of the single-dimensional gradients in (84). Furthermore, as in Chapter 4, when the allocation decisions are at equidistance from the locations, then (91) and (92) would be zero since (93) and (94) will be zero. We now provide the formal description of the hybrid improvement algorithm.

Hybrid Improvement Algorithm:

Step 1. Define and Initialize the model parameters and variables

j : index for optimality iterations (i.e. $j^* = \{j| \ \epsilon_{COST} \geq \frac{|TC^{j+1}-TC^{j}|}{|TC^{j}|}\}$

ϵ_{COST} : epsilon parameter for optimality stopping decision

ϵ_{SLA} : epsilon parameter for skipping the steepest-descent improvement step and performing one iteration of SLA

α^j : step length for line search iterations at the j^{th} iteration

α_{TEST} : step length for comparison with SLA approach and Steepest-descent improvement solution

$\mathcal{A}^j_{i=1,2,..,n}$: set of allocation decisions at iteration j

$(\mathbf{x}^*_i)^j = (x^*_i, y^*_i)^j$: optimal locations corresponding to $\mathcal{A}^j_{i=1,2,..,n}$

M : market boundary parameter (assuming a square-shaped market region)

Set $j = 0$

Step 2. Initialization: Allocate the service regions and optimally locate facilities

Step 2.1. Generate the an initial set of voronoi-generator points $p_{i=1,2,...,n}$

Step 2.2. Generate the voronoi-diagram of the $p_{i=1,2,...,n}$ and identify the following sets:

Allocation polygons $\mathcal{A}^j_i$,

Step 2.3. Find the optimal locations and calculate the total cost.

$$(\mathbf{x}_i^*)^j = (x_i^*, y_i^*)^j := \arg \min_{(x_i,y_i)} (\int_{\mathcal{A}_i} d_p(\mathbf{x}_i, \mathbf{x}) D(\mathbf{x}) d\mathbf{x})$$
$$TC^j = \sum_{i=1,2,\ldots n} \int_{\mathcal{A}_i^j} d_p((\mathbf{x}_i^*)^j, \mathbf{x}) D(\mathbf{x}) d\mathbf{x}$$

Step 3. Check for optimality

Step 3.1. Optimality check

If $j > 1$ and $\epsilon_{COST} \geq \frac{|TC^{j+1} - TC^j|}{|TC^j|}$, Go to **Step 6.**

Else set $j := j + 1$ and continue with **Step 3.2.**

Step 3.2. Calculate the partial gradients

For each optimal location $(\mathbf{x}_i^*)^j$, repeat :

1- Find partial derivative of Single-dimensional allocation decisions $(A_i^x(y), A_i^y(x))$ with respect to $(x_i^*, y_i^*)^j$

$\frac{\partial A_i^x(y)}{\partial x_i^*}$ and $\frac{\partial A_i^y(x)}{\partial y_i^*}$

2- Find partial derivative of TC with respect to Single-dimensional allocation decisions $(A_i^x(y), A_i^y(x))$

$\frac{\partial TC}{\partial A_i^x(y)}$and $\frac{\partial TC}{\partial A_i^y(x)}$ using (93) and (94)

3- Find partial derivative of TC with respect to $(\mathbf{x}_i^*)^j$,i.e. $\frac{\partial TC}{\partial (\mathbf{x}_i^*)^j}$

$$\frac{\partial TC}{\partial x_i^*} = \int_{y \in Y_{BR}} \left(\frac{\partial TC}{\partial A_{i-1}^x(y)} \frac{\partial A_{i-1}^x(y)}{\partial x_i^*} + \frac{\partial TC}{\partial A_i^x(y)} \frac{\partial A_i^x(y)}{\partial x_i^*} \right) dy$$

$$\frac{\partial TC}{\partial x_i^*} = \int_{y \in Y_{BR}} \left(\frac{\partial TC}{\partial A_{i-1}^x(y)} \frac{\partial A_{i-1}^x(y)}{\partial x_i^*} + \frac{\partial TC}{\partial A_i^x(y)} \frac{\partial A_i^x(y)}{\partial x_i^*} \right) dy$$

Normalize the gradients $\mathbf{d}_{\mathbf{x}_i^*}$ to obtain gradient vectors, $\mathbf{d}_{\mathbf{x}_i^*}^{norm}$, for $\forall\, (\mathbf{x}_i^*)^j$

Step 4. Compare SLA performance with the improvement direction

Step 4.1. SLA Approach

· **Solve LAM-AVS by relaxing constraint (69)**

Find Nearest-Neighbor solution for $(\mathbf{x}_i^*)^j$

Assign solution to $(\mathcal{A}_i)^j_{SLA}$

· **Solve LAM-LVS by relaxing constraint (66)**

Solve Single-facility problem in $(\mathcal{A}_i)^j_{SLA}$

Assign solution to $(\mathbf{x}_i^*)^j_{SLA}$

· **Calculate total cost**

$$TC^j_{SLA} = \sum_{i=1,2,\dots n} \int_{(\mathcal{A}_i)^j_{SLA}} d_p((\mathbf{x}_i^*)^j_{SLA}, \mathbf{x}) D(\mathbf{x}) d\mathbf{x}$$

Step 4.2. Improvement Approach

· **Update the optimal location decisions**

$(\mathbf{x}_i^*)^j_{DESCENT} := (\mathbf{x}_i^*)^j - \alpha_{TEST}\, \mathbf{d}^{norm}_{\mathbf{x}_i^*}$

· **Solve LAM-AVS by relaxing constraint (69)**

Find Nearest-Neighbor solution for $(\mathbf{x}_i^*)^j_{DESCENT}$

Assign solution to $(\mathcal{A}_i)^j_{DESCENT}$

· **Solve LAM-LVS by relaxing constraint (66)**

Solve Single-facility problem in $(\mathcal{A}_i)^j_{DESCENT}$

Assign solution to $(\mathbf{x}_i^*)^j_{DESCENT}$

· **Calculate total cost**

$$TC^j_{DESCENT} = \sum_{i=1,2,\dots n} \int_{(\mathcal{A}_i)^j_{DESCENT}} d_p((\mathbf{x}_i^*)^j_{DESCENT}, \mathbf{x}) D(\mathbf{x}) d\mathbf{x}$$

Step 4.3. Compare $TC^j_{DESCENT}$ and TC^j_{SLA}

If $\frac{TC^j_{DESCENT} - TC^j_{SLA}}{TC^j_{DESCENT}} \geq \epsilon_{SLA}$, then Go to **Step 5**,

else assign

$(\mathbf{x}_i^*)^{j+1} := (\mathbf{x}_i^*)^j_{SLA}$ and $(\mathcal{A}_i)^{j+1} := (\mathcal{A}_i)^j_{SLA}$, and $TC^j := TC^j_{SLA}$

Return to **Step 3.**

Step 5. Steepest-Descent Improvement Line search

Step 5.1. Repeat following until $(\alpha^j)^* := \arg\min_{\alpha^j} TC'$:

· **Update the optimal location decisions**

$(\mathbf{x}_i^*)' := (\mathbf{x}_i^*)^j - \alpha^j \mathbf{d}_{\mathbf{x}_i^*}^{norm}$

· **Solve LAM-AVS by relaxing constraint (69)**

Find Nearest-Neighbor solution for $(\mathbf{x}_i^*)'$

Assign solution to $(\mathcal{A}_i)'$

· **Solve LAM-LVS by relaxing constraint (66)**

Solve Single-facility problem in $(\mathcal{A}_i)'$

Assign solution to $(\mathbf{x}_i^*)'$

· **Calculate total cost**

$TC' = \sum_{i=1,2,\ldots n} \int_{(\mathcal{A}_i)'} d_p((\mathbf{x}_i^*)', \mathbf{x}) D(\mathbf{x}) d\mathbf{x}$

Step 5.2. Assign $(\mathbf{x}_i^*)^{j+1} := (\mathbf{x}_i^*)'$, $(\mathcal{A}_i)^{j+1} := (\mathcal{A}_i)'$, $TC^j := TC'$ and Return **Step 3.**

Step 6. Terminate with the optimal solution $\mathcal{A}_i^j$ and $\mathbf{x}_i^*$

We now provide an illustrative example of the above algorithm.

Example 7.4: Hybrid Improvement Algorithm - Manhattan-metric (L_1) Case

Let's consider an example implementation of the above algorithm on a small scale example of 3-facility for the Manhattan-metric case. Our market region is a square market region $\mathcal{M}$=$\{(x, y)|x \in (0, 100)$ and $y \in (0, 100)\}$, i.e. $M = 100$. and the demand density function is a linear density function ($D(x, y) = 100 + 10x + 5y$) over the market region $\mathcal{M}$.

ITERATION 1

Step 1. Define and Initialize the model parameters and variables

Set $j = 0$

$\epsilon_{COST} := 5 \times 10^{-4}$

$\epsilon_{SLA} := 1 \times 10^{-7}$

$\alpha_{TEST} := 0.01$

Step 2. Initialization: Allocate the service regions and optimally locate facilities

Step 2.1. Initial generator points: $p_1 = (21, 52)$, $p_2 = (21, 52)$, $p_1 = (21, 52)$

Step 2.2. Generate the voronoi-diagram of the $p_{i=1,2,3}$ and identify the allocation polygons $\mathcal{A}_{i=1,2,3}^{j=0}$

Step 2.3. Find the optimal locations and calculate the total cost.

$(\mathbf{x}_1^*)^{j=0} = (22.116, 49.567), (\mathbf{x}_2^*)^{j=0} = (63.467, 83.106),$
$(\mathbf{x}_3^*)^{j=0} = (80.261, 33.687)$

$TC^{j=0} = 247, 641, 899.7$

$(\mathbf{x}_i^*)^{j=0}$ and $\mathcal{A}_i^{j=0}$ are displayed in Figure 7.19.

Step 3. Check for optimality

Step 3.1. Optimality check

$j := 1$

Step 3.2. Calculate the partial gradients

$\mathbf{d}_{\mathbf{x}_1^*} = (-796, 742.1, -433, 628.1)$

$\mathbf{d}_{\mathbf{x}_2^*} = (-1, 620, 312.2, 367, 934.9)$

$\mathbf{d}_{\mathbf{x}_3^*} = (-369, 543.5, 855, 000.9)$

Normalize the gradients $\mathbf{d}_{\mathbf{x}_i^*}$ to obtain gradient vectors, $\mathbf{d}_{\mathbf{x}_i^*}^{norm}$, for $\forall\, (\mathbf{x}_i^*)^j$

$\mathbf{d}_{\mathbf{x}_1^*}^{norm} = (-0.3776407091, -0.2055315576)$

$\mathbf{d}_{\mathbf{x}_2^*}^{norm} = (-0.7679974108, 0.1743942199)$

$\mathbf{d}_{\mathbf{x}_3^*}^{norm} = (-0.1751566286, 0.4052543186)$

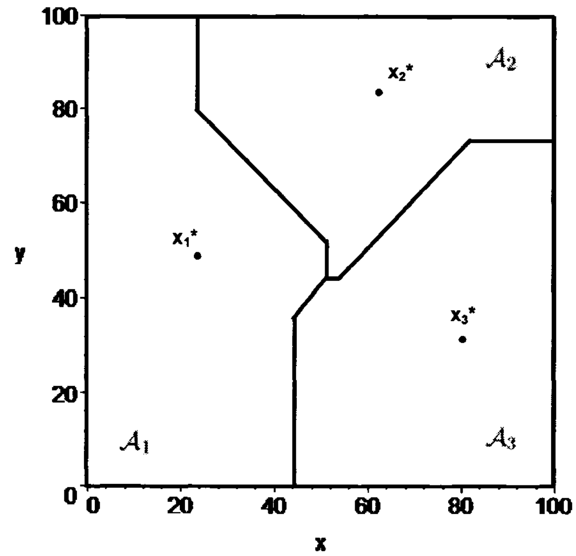

Figure 7.19: Starting solution for the example implementation of the Hybrid Improvement Algorithm for Manhattan-metric (Example 7.4).

Step 4. Compare SLA performance with the improvement direction

Step 4.1. SLA Approach

$(\mathbf{x}_1^*)_{SLA}^{j=1} = (24.155, 52.006), (\mathbf{x}_2^*)^{j=1} = (67.803, 82.477),$
$(\mathbf{x}_3^*)^{j=1} = (79.772, 31.052)$

$(\mathbf{x}_i^*)_{SLA}^{j=1}$ and $(\mathcal{A}_i)_{SLA}^{j=1}$ are displayed in Figure 7.20.

· **Calculate total cost**

$TC_{SLA}^{j=1}$ =242,830,415.8

Step 4.2. Improvement Approach

· **Update the optimal location decisions**

$(\mathbf{x}_i^*)_{DESCENT}^{j=1} := (\mathbf{x}_i^*)^{j=1} - 0.01\ \mathbf{d}_{\mathbf{x}_i^*}^{norm}$

$(\mathbf{x}_1^*)_{DESCENT}^{j=1} = (24.158, 52.008), \quad (\mathbf{x}_2^*)^{j=1} = (67.808, 82.476),$
$(\mathbf{x}_3^*)^{j=1} = (79.772, 31.048)$

· **Solve LAM-AVS by relaxing constraint (69)**

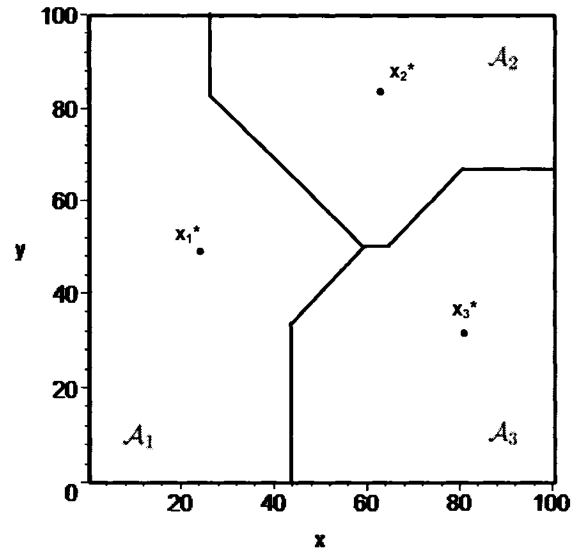

Figure 7.20: Solution from direct implementation of the SAL in the first iteration (Example 7.4).

· **Solve LAM-LVS by relaxing constraint (66)**

· **Calculate total cost**

$TC_{DESCENT}^{j=1} = 242,825,846.60$

Step 4.3. Compare $TC_{DESCENT}^{j=1}$ and $TC_{SLA}^{j=1}$

$\frac{TC_{DESCENT}^{j=1} - TC_{SLA}^{j=1}}{TC_{DESCENT}^{j=1}} = 188.17 \times 10^{-7} \geq \epsilon_{SLA} = 1 \times 10^{-7}$, thus continue with **Step 5**,

Step 5. Steepest-Descent Improvement Line search

· $(\alpha^{j=1})^* := \arg\min_{\alpha^{j=1}} TC' = 15.6$

$(\mathbf{x}_1^*)' := (28.532, 55.775), (\mathbf{x}_2^*)' := (74.695, 80.462)$,

$(\mathbf{x}_3^*)' := (79.662, 26.595)$

$(\mathbf{x}_i^*)^2 := (\mathbf{x}_i^*)'$ and $(\mathcal{A}_i)^2 := (\mathcal{A}_i)\prime$ are illustrated in Figure and corresponding total cost is $TC^1 := TC' = 238,698,979.4$

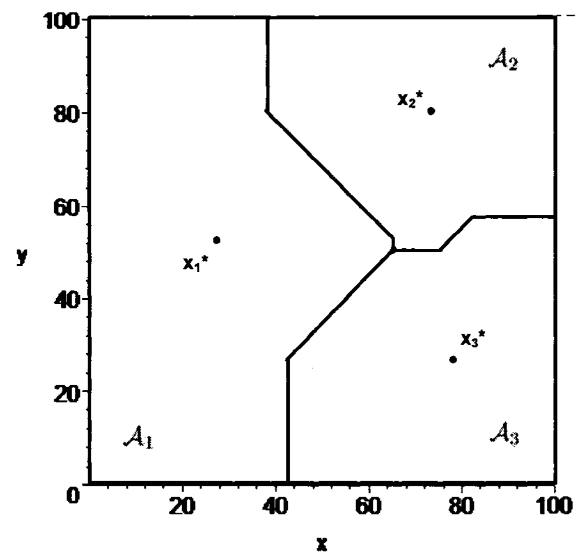

Figure 7.21: Solution in the end of first iteration of the Hybrid Improvement Algorithm for Manhattan-metric (Example 7.4).

ITERATION 2

Step 3. Check for optimality

Step 3.1. Optimality check

$j := 2$

Step 3.2. Calculate the partial gradients

$\mathbf{d}_{\mathbf{x}_1^*} = (116, 454.14, -333, 575.06)$

$\mathbf{d}_{\mathbf{x}_2^*} = (-252, 388.02, -65, 570.80)$

$\mathbf{d}_{\mathbf{x}_3^*} = (629, 593.06, -114, 993.85)$

Normalize the gradients $\mathbf{d}_{\mathbf{x}_i^*}$ to obtain gradient vectors, $\mathbf{d}_{\mathbf{x}_i^*}^{norm}$, for $\forall\, (\mathbf{x}_i^*)^{j=2}$

$\mathbf{d}_{\mathbf{x}_1^*}^{norm} = (0.1500363677, -0.4297690997)$

$\mathbf{d}_{\mathbf{x}_2^*}^{norm} = (-0.3251698989, -0.08447964223)$

$\mathbf{d}_{\mathbf{x}_3^*}^{norm} = (0.8111506895, -0.1481549699)$

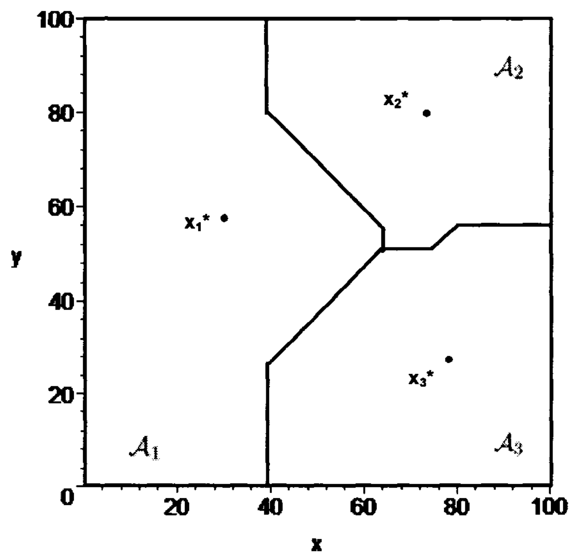

Figure 7.22: Solution from direct implementation of the SAL in the second iteration (Example 7.4).

Step 4. Compare SLA performance with the improvement direction

Step 4.1. SLA Approach

$(\mathbf{x}_1^*)_{SLA}^{j=2} = (28.169, 57.616), (\mathbf{x}_2^*)^{j=2} = (75.458, 80.380),$

$(\mathbf{x}_3^*)^{j=2} = (78.198, 26.436)$

$(\mathbf{x}_i^*)_{SLA}^{j=2}$ and $(\mathcal{A}_i)_{SLA}^{j=2}$ are displayed in Figure 7.22.

· **Calculate total cost**

$TC_{SLA}^{j=2} = 237,911,402.0$

Step 4.2. Improvement Approach

· **Update the optimal location decisions**

$(\mathbf{x}_i^*)_{DESCENT}^{j=2} := (\mathbf{x}_i^*)^{j=2} - 0.01\ \mathbf{d}_{\mathbf{x}_i^*}^{norm}$

$(\mathbf{x}_1^*)_{DESCENT}^{j=2} = (28.168, 57.620), (\mathbf{x}_2^*)^{j=2} = (75.461, 80.3798),$

$(\mathbf{x}_3^*)^{j=2} = (78.194, 26.437)$

· **Solve LAM-AVS by relaxing constraint (69)**

· **Solve LAM-LVS by relaxing constraint (66)**

· **Calculate total cost**

$TC^{j=2}_{DESCENT} = 237,909,774.60$

Step 4.3. Compare $TC^{j=2}_{DESCENT}$ and $TC^{j=2}_{SLA}$

$\frac{TC^{j=2}_{DESCENT} - TC^{j=2}_{SLA}}{TC^{j=2}_{DESCENT}} = 68.4 \times 10^{-7} \geq \epsilon_{SLA} = 1 \times 10^{-7}$, thus continue with **Step 5**,

Step 5. Steepest-Descent Improvement Line search

· $(\alpha^{j=2})^* := \arg\min_{\alpha^{j=2}} TC' = 5.75$

$(\mathbf{x}_1^*)' := (27.575, 60.059), (\mathbf{x}_2^*)' := (77.092, 80.339), (\mathbf{x}_3^*)' := (75.702, 26.628)$

$(\mathbf{x}_i^*)^{j=3} := (\mathbf{x}_i^*)'$ and $(\mathcal{A}_i)^{j=3} := (\mathcal{A}_i)'$ are illustrated in Figure 7.23 and corresponding total cost is $TC^{j=2} := TC' = 237,292,546.7$

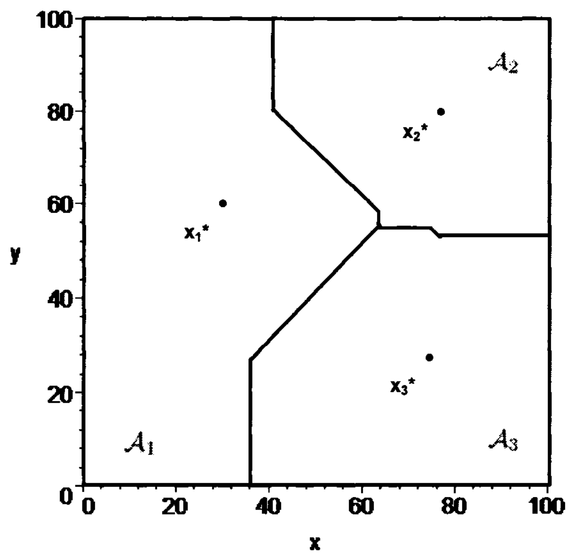

Figure 7.23: Solution in the end of second iteration of the Hybrid Improvement Algorithm for Manhattan-metric (Example 7.4).

When we continue with two more iterations, we reach the optimality specified by $\epsilon_{COST} = 5 \times 10^{-4}$. The results are summarized in the Table 7.5. Third column in the table, TC_{SLA}, represents the solution obtained when we implement a direct SAL iteration instead of the hybrid-algorithm. Accordingly, the last column presents the factor by which performing a line search in the descent direction improves the solution over that of direct implementation of the SAL. Even though, by design, the hybrid approach's worst case performance is equivalent to the SLA (or SAL) method, last column shows that it is indeed better.

Iter. No	TC*	TC_{SLA}	Improvement over SLA
0	247,641,899.70		
1	238,698,979.40	242,830,415.80	1.86
2	237,292,546.70	237,911,402.00	1.79
3	237,067,825.20	237,151,460.80	1.59
4	237,024,382.70	237,043,355.30	1.78

Table 7.5: Iteration results for the Hybrid algorithm for the Manhattan-metric (L_1) example (Example 7.4).

The resulting solution is as follows and illustrated in Figure 7.24.

$$(\mathbf{x}_1^*)^{j=5} = (27.831, 62.0), (\mathbf{x}_2^*)^{j=5} = (78.931, 79.538), (\mathbf{x}_3^*)^{j=5} = (73.920, 26.009)$$

$$TC^{j=5} = 237,024,382.70$$

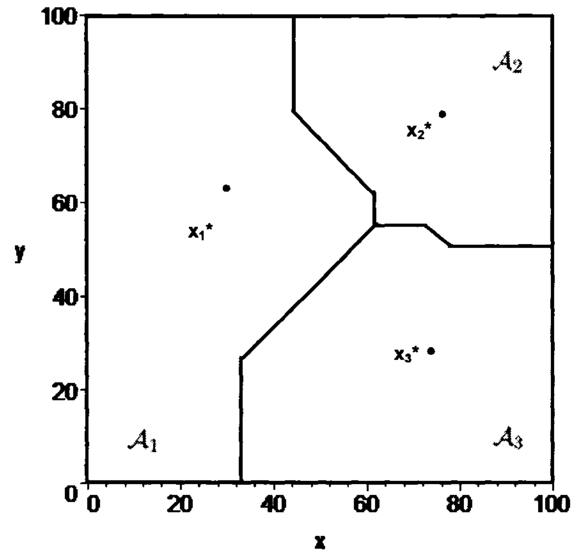

Figure 7.24: Solution in the end of fourth iteration of the Hybrid Improvement Algorithm for Manhattan-metric (Example 7.4).

7.5 Conclusions

This chapter presents alternative modeling and solution techniques for the location-allocation problems based on the Manhattan-metric. There are two main contributions of this chapter. Firstly, we have extended the two main classes of solution methods, constructive and improvement-based techniques, to the planar 2-facility problems. Secondly, we have developed a hybrid-algorithm which allows us to solve the location-allocation problems in the allocation variable space.

In the first contribution, we have extended the definition of the allocation line from a straight line form for the Euclidean-metric based distance measures to a three-segment form for the Manhattan-metric case. With this revision, we have adapted both the constructive solution approach and improvement-based steepest-descent algorithms to the planar 2-facility case based on the Manhattan metric. This contribution hints that, when the particular shape of the optimal allocation decisions are known, we could adopt these two main classes of solution methods to any another metric. However, as the form of the optimal allocation decision becomes more complex, i.e. higher degrees of freedom, then the solution performance of these methods degrades.

The second contribution is the hybrid improvement approach, which is a composition mapping of the steepest-descent and sequential location-allocation (SLA) method. This approach is superior to the vertex-iteration based approach as it is not limited by the assumption of shape invariant allocation decisions. In this approach, we are able to combine the solution-improvement strength of the descent search methods with the simple, yet good-performing aspect of the SLA method. The closeness and global convergence of this algorithm follows from Zangwill's theorems on the closeness of the algorithmic compositions (Luenberger 1984).[27] Although, this approach is primarily developed for the Manhattan-metric due to the generalizable design of the algorithm, i.e. the composition of the steepest-descent for optimal location iterations and one step SLA, this algorithm can be adapted to a variety of different metrics with a guaranteed worst-case performance of the SLA approach. The main ingredient for the implementation of this hybrid algorithm is the derivation of the differential relationship between the single dimensional allocation decisions and optimal locations for a given allocation solution. The only drawback to this hybrid approach is the inability to accommodate approximate second-order procedures such as conjugate-gradient or Quasi-Newton methods.

[27] By "global convergence" we referring to the convergence property of this composite algorithm from any starting solution to a local optimum solution in finite number of iterations.

Chapter 8

Conclusions and Future Research

In this chapter we present a conclusion of our results and findings, describe our contribution to the literature and state future research directions.

8.1 Concluding Remarks

In this dissertation, we have developed an allocation-based modeling and solution framework for location-allocation problems with continuous demand data. Our objectives have been two fold. The first objective is to address the complexity issue of the problems when the demand is dense. The second objective is to provide a framework based on the allocation decisions such that our model and solution approach can account for more general problem characteristics, i.e. constrained problems and capacity dependent costs.

We have primarily used the single dimensional setting in Chapter 3, i.e. line, to develop and test our algorithms which can be extended to planar settings. One observation we had is that, planar problem extension of allocation space models, is remarkably difficult due to the topological properties of planar allocation decisions. In the case of location based models, this transition is not as difficult, since on both the line and the plane, points are used to characterize the locations. In comparison, while allocation decisions on the line are line segments, on the plane, allocation decisions are areas. In order to accommodate this difference, we have differentiated between two metrics, Euclidean and Manhattan metrics. This differentiation is necessary, because the constructs used to define their allocation decisions on the plane are dissimilar. For the Euclidean metric, straight lines are the separators of allocation decisions, whereas Manhattan metric separation is via a piecewise linear form.

In the 2-facility Euclidean-metric problems, Chapter 4, a straight line is sufficient to characterize the allocation decisions. This has been also observed for the discrete demand cases by Francis and White (1998), Ostresh (1975), Drezner (1984) and O'Kelly (1986). However, we have implemented the continuous demand versions and developed two solution procedures. The first solution procedure is the constructive approach where we solve first-order conditions by relaxing the boundary constraints. Second approach is an improvement based method, where, starting with an initial solution, we change the slope and intercept of the allocation line using first or second order gradient information to improve the solution quality. In the case of the Manhattan-metric 2-facility problems, Chapter 7, the constructive solution approach is similar except that we need to solve one more first-order condition. The improvement approach for the Manhattan-metric uses the special structure of the allocation line and independently translates segments of the allocation line.

In the n-facility Euclidean-metric case, Chapter 5, we have developed an improvement based solution procedure where we move the vertices of the allocation polygons. This approach uses the results developed for the 2-facility case to calculate the effect on the edges connecting to the vertices and aggregates this information at the vertex level. In a sense we move from the slope and intercept decision variable space of the 2-facility case to vertex variable space in the n-facility case. Since we iterate the vertices which are connected to each other, some iterations results in infeasible allocation solutions, i.e. overlapping allocation decision polygons. We account for these infeasibilities by the vertex-event handling procedures as described in Chapter 5. Specifically, with these vertex-event handling procedures, we recover the feasibility of the allocation solutions while allowing solution improving movements as much as possible. In order to understand the runtime performance of our algorithm (i.e. steepest-descent method based on vertex iteration) as well as to gain insights into the effect of problem parameters on the solution, we perform a computational study in Chapter 6.

For the n-facility Manhattan-metric case, Chapter 7, we note that the vertex iteration approach is not applicable for the Manhattan-metric due to the shape of the allocation line. Accordingly, we developed a hybrid algorithm which combines one step of the sequential location-allocation (SLA) method of Cooper (1964) with the descent based improvement approach. With this procedure, we are able to solve the problem in the allocation variable space with a guaranteed performance which is better than the steepest descent or SLA approach alone. However, this approach requires calculation of the gradients for the optimal locations which are surrogate iterates of the allocation decisions. Note that this approach is generalizable to other metrics as well.

8.2 Contributions to Research

Our contribution to the literature with this thesis can be classified into four categories:

1- Development of a modeling and solution framework for continuous demand planar location-allocation problems based on the allocation decisions

In real applications of the location-allocation problems the demand data is usually large (Taillard 2003, Brimberg 2000, Miller 1996). One way to overcome this is the aggregation approach, which results in various types of errors (Erkut and Bozkaya 1999, Norman et al. 1999). Accordingly, continuous demand, a form of disaggregation, provides two benefits. Firstly, it is more accurate and accessible with current technologies (Miller 1996, Drezner 1997). Second, the continuous demand smooths the objective function; thus, local optimization techniques have higher a chance to converge to a global solution Drezner (1997). Furthermore, traditional approaches focus on location decisions and assume optimal allocation decisions given the locations. Accordingly allocation decisions are dependent on the location decisions. Choice of locations involves consideration of many different factors other than the trade-offs considered in location-allocation problems. Hence, with this thesis, we complement the location-based approaches by providing a framework which could defer the site selection decisions until after the allocation decisions (service regions) are made. Since our model and solution procedure operate on continuous demand, our approach also provides such benefits as smoothing of the objective function and avoiding aggregation errors.

Continuous approximation literature for locational-allocation problems assumes the demand is slowly varying and that service regions are in the form of certain shapes (i.e. round). However, this is a restrictive assumption and could lead to significant errors in the location and allocation decisions (Chapter 3 and Chapter 6 in this thesis). On the other hand continuous approximation is favorable due to its reduced data requirements. With our approach, we contribute to the literature by providing the same desirable property of reduced data requirements without having any restrictions on the demand or the service regions.

2- Development of an efficient solution approach for 2-facility problems

Francis and White (1998), Ostresh (1975), Drezner (1984) and O'Kelly (1986) utilized the convex hull property of optimal subsets that they must be separated by a line. However, these methods are limited to small size problems due to the increasing number of such partitions. When we consider a continuous demand function as an infinite number of demand points, surely this property is not useful. Instead we utilize the same line separation property and develop computationally efficient constructive solution approaches for planar 2-facility problems. This approach is based on solving a first-order condition which jointly characterizes optimal location and allocation decisions. For the manhattan-metric, line separation is different than the straight line, thus we have adjusted the constructive solution procedure accordingly. The constructive solution property relies on the equidistance of the allocation line to the optimal locations. This might not be the case for different settings, such as nonlinear capacity acquisition costs. Hence, we also developed an gradient based improvement solution approach which can handle these cases.

3- Develop an alternative approach to the voronoi-based solution approaches for Euclidean-metric location-allocation problems with continuous demand

The voronoi diagram approach for solving location-allocation problems with continuos demand involves iteration of the location decisions (Iri et al. 1983). At each step, a new voronoi diagram is constructed based on the new locations. In this thesis, we show that it is possible to solve Euclidean-metric location-allocation problems without the need of reconstructing the voronoi diagram, except at the start to obtain an initial allocation solution. Specifically, we provide a vertex-iteration based update of the service region districts. In this approach vertices characterizing the allocation polygons are iterated. In the case of infeasibility, such as overlapping allocation polygons, we developed a set of feasibility recovery procedures called vertex-event handling procedures.

4- Develop a hybrid improvement algorithm for Manhattan-metric location-allocation problems

The sequential location-allocation solution approach is a popular method for solving location-allocation problems. This approach, though simple and devoid of line-search step, is known to be slow in its convergence rate (Taillard 2003, Brimberg et al. 2000). We develop a hybrid improvement approach, i.e. composition of steepest-descent with the SLA method, which provides an improvement over sole implementation of either method. One major aspect of this approach is that it can be generalized to other metrics and still solve the problem in the allocation variable space. Accordingly, the flexibility associated with the allocation space (contribution 1) can be extended to the Manhattan as well as to other metrics.

8.3 Future Research

There are a number of possible extensions of this research.

1. Comparison of the error bound with demand aggregation versus demand disaggregation

For the most part, the location-allocation literature assumes discrete demand data. Studies with continuous demand are scarce (Fekete, 2005), except the continuos approximation literature which primarily assumes a continuous demand. Hence, there is a need for further investigation of the error types and bounds when we disaggregate the discrete demand data. Since the aggregation literature is well developed in this direction, similar steps can be taken to investigate the pros/cons of replacing discrete demand with continuos demand density function. Furthermore, comparison the effect of these two alternative methods (aggregation vs. disaggregation) on the solution of location-allocation problems would be the next step.

2. Extension of the vertex-based improvement method to constrained problems

In the location-allocation problems several studies considered such constraints as limited capacity, restricted distances and forbidden regions for locating facilities. We plan to incorporate these model features and adapt a barrier or penalty based method to solve these constrained problems. Limited capacity can be handled with our model much easier than the alternative modeling approach in the location variable space. Location based models iterate locations and then perform a voronoi tesselation (i.e. nearest assignment) during which accounting for the capacity restrictions is not possible. In our approach, we iterate the allocation decisions on a smooth subspace thus we could control or price them.

3. Extension of the vertex-based improvement method to account for location and allocation dependent cost parameters

Throughout this thesis we assumed fixed costs, capacity acquisition costs and transportation costs are independent of the location and allocation amount. In other words, every unit of demand cost us the same in unit distance transportation and capacity acquisition. However, the model would be more practicable if we could express them as spatial variables and include economies of scale in the capacity acquisition cost (Berman and Parkan 1984, Drezner and Wesolowsky 1989, 1999a, Berman and Drezner 2002, Brimberg and Salhi 2005). Furthermore, by assigning very large fixed costs to the regions where facility location is not allowed, we could solve the location-allocation problems with forbidden regions (Fliege and Nickel 2000).

4. Competitive facility location

In the competitive models, competition takes places over the catchment areas, i.e. allocation decisions (Dasci and Laporte 2005b, Berman and Krass 1992, Drezner 1982). Hence, since we model and solve the problem in the allocation variable space, it is possible to adopt our approach for competitive problems.

9 Appendices

9.1 Appendix 3

Proposition 3.1

For any given triplet of service regions (A_i, A_{i+1}, A_{i+2}), $TC_{i,3}(B_i, A_i)\{= TC(B_i, A_i) + TC(B_{i+1}, A_{i+1}) + TC(B_{i+2}, A_{i+2}),\ s.t.\ B_{j+1} = B_j + A_j\ for\ j = i, i+1, i+2\}$ *is a strongly quasi-convex function of* A_i *and* A_{i+1} *for a given* B_i *and* B_{i+3}.

Proof.

We will prove this proposition in two major steps. In the first step, we focus on the total cost of last two service regions and prove its quasi-convexity for A_{i+1}. Then, in step 2, we focus on the terminal service region triplet and prove quasi-convexity of its total cost in A_{i+1} and B_{i+1}. Since fixed costs are independent allocation decisions, we exclude them from consideration. In addition, the total capacity acquisition cost in a given market region is constant with linear capacity acquisition cost, thus we also ignore them for notational simplicity.[28]

STEP 1.

First we prove that $TC_{n-1,2}(B_{n-1}, A_{n-1})$ is a strongly quasi-convex function of A_{n-1} for any given B_{n-1}. (note that $B_n = B_{n-1} + A_{n-1}$ and $M = B_n + A_n$)

$$TC_{n-1,2}(B_{n-1}, A_{n-1}) = TC(B_{n-1}, A_{n-1}) + TC(B_n, A_n),\ where\ B_n = M - A_n$$

[28]Total capacity acquisition cost is indepedent of the allocation decisions since $\sum_{i=1}^{n}\left(\int_{A_i} D(x)dx\right) = \int_M D(x)dx$.

a) $TC(B_{n-1}, A_{n-1})$ is a non-decreasing (i) and convex (ii) function of A_{n-1} for any given B_{n-1}.

i) Show that $TC(B_{n-1}, A_{n-1})$ is **non-decreasing** in A_{n-1} for any given B_{n-1}:

It can be shown by the first order condition

$$\frac{\partial TC(B_{n-1}, A_{n-1})}{\partial A_{n-1}} = \frac{\partial TC(B_{n-1}, A_{n-1})}{\partial T_{n-1}} \frac{\partial T_{n-1}}{\partial A_{n-1}} \geq 0$$

where

$$\begin{aligned} & \frac{\partial TC(A_{n-1}, B_{n-1})}{\partial T_{n-1}} \\ = \; & \frac{c(u + vB_{n-1})^3}{2v^2} \left(\frac{2G(T_{n-1})(T_{n-1}+1)^2 - \sqrt{2}G(T_{n-1})^2(T_{n-1}+1)}{G(T_{n-1})} \right) \end{aligned}$$

$$T_{n-1} = \frac{vA_{n-1}}{u + vB_{n-1}}$$

$$G(T_{n-1}) = \sqrt{T_{n-1}^2 + 2T_{n-1} + 2}$$

$$\frac{\partial T_{n-1}}{\partial A_{n-1}} = \frac{v}{u + vB_{n-1}}$$

It can be easily verified that $\frac{\partial TC(B_{n-1}, A_{n-1})}{\partial T_{n-1}} \geq 0 \; (\leq 0)$ and $\frac{\partial T_{n-1}}{\partial A_{n-1}} \geq 0 \; (\leq 0)$ when $v \geq 0 \; (\leq 0)$, thus $\frac{\partial TC(B_{n-1}, A_{n-1})}{\partial A_{n-1}} \geq 0$ when $v \gtreqless 0$.

ii) Show that $TC(B_{n-1}, A_{n-1})$ is **convex** in A_{n-1} for any given B_{n-1}:

$TC(B_{n-1}, A_{n-1})$ is convex if second order derivative is nonnegative. However we will use a much more easier approach in showing that $TC(B_{n-1}, A_{n-1})$ is convex using the composition conditions of convexity.

For $TC(B_{n-1}, A_{n-1})$ to be convex either:

1. $T_{n-1}(A_{n-1}) = \frac{vA_{n-1}}{u+vB_{n-1}}$ is convex in A_{n-1} and $TC(B_{n-1}, A_{n-1})$ is convex and non-decreasing in T_{n-1}. $(v \geq 0)$

$T_{n-1}(A_{n-1})$ is obviously convex and further from the results of part (i) $TC(B_{n-1}, A_{n-1})$ is non-decreasing in T_{n-1}. What remains to be shown is $TC(B_{n-1}, A_{n-1})$ is convex in T_{n-1}.

$$\begin{aligned} & \frac{\partial^2 TC(B_{n-1}, A_{n-1})}{\partial T_{n-1}^2} \\ = \; & \frac{c(u+vB_{n-1})^3}{2v^2}\left(\frac{4G(T_{n-1})(T_{n-1}+1) - 2\sqrt{2}G(T_{n-1})^2 + \sqrt{2}}{G(T_{n-1})}\right) \end{aligned}$$

and since $4(T_{n-1}+1) - 2\sqrt{2}G(T_{n-1}) \geq 0$ (even for $v \leq 0$), then $TC(B_{n-1}, A_{n-1})$ is convex in T_{n-1}. Thus $TC(B_{n-1}, A_{n-1})$ is convex in A_{n-1} when $v \geq 0$.

2. $T_{n-1}(A_{n-1}) = \frac{vA_{n-1}}{u+vB_{n-1}}$ is concave in A_{n-1} and $TC(B_{n-1}, A_{n-1})$ is convex and non-increasing in T_{n-1}. $(v \leq 0)$

$T_{n-1}(A_{n-1})$ is also concave and from the results of case when $v \geq 0$, $TC(B_{n-1}, A_{n-1})$ is non-decreasing in T_{n-1} and $TC(B_{n-1}, A_{n-1})$ is convex in T_{n-1}. Thus $TC(B_{n-1}, A_{n-1})$ is convex in A_{n-1} when $v \leq 0$. Therefore $TC(B_{n-1}, A_{n-1})$ is convex in A_{n-1} when $v \gtreqless 0$.

b) $TC(B_n, A_n)$ is a strongly quasi-convex and non-increasing function of A_{n-1} for any given B_{n-1}, where $B_n = M - A_n$ and $A_n = M - B_{n-1} - A_{n-1}$.

First, we show that $TC(B_n, A_n)$ is a non-decreasing function of A_{n-1} (i.e. $\frac{\partial TC(B_n, A_n)}{\partial A_{n-1}} \leq 0$)

Define $T_n = \frac{vA_n}{u+vB_n} = \frac{v(M-B_{n-1}-A_{n-1})}{u+vB_{n-1}+vA_{n-1}}$, then we need to show:

$$\frac{\partial TC(B_n, A_n)}{\partial A_{n-1}} = \frac{\partial TC(B_n, A_n)}{\partial T_n}\frac{\partial T_n}{\partial A_{n-1}} \leq 0$$

$$\frac{\partial TC(B_n, A_n)}{\partial T_n} = \frac{c(u+vM)^3}{2v^2(T_n+1)^4}\left(\frac{\sqrt{2}G(T_n)^2 - 2G(T_n)}{G(T_n)}\right)$$

$$\frac{\partial T_n}{\partial A_{n-1}} = \frac{-v(T_n+1)^2}{u+vM}$$

It can be easily verified that $\frac{\partial TC(B_n,A_n)}{\partial T_n} \geq 0$ $(\leq 0)$and $\frac{\partial T_n}{\partial A_{n-1}} \leq 0$ (≥ 0) when $v \geq 0$ (≤ 0), thus $\frac{\partial TC(B_n,A_n)}{\partial A_{n-1}} \leq 0$ when $v \gtreqless 0$ and $TC(B_n, A_n)$ is a strictly non-increasing therefore strongly quasi-convex function of A_{n-1}.

Next we need to further explore the particular behavior of $TC(B_n, A_n)$ over A_{n-1} in order to characterize the result of its summation with $TC(B_{n-1}, A_{n-1})$.

(i) When $v \geq 0$, $\frac{\partial TC(B_n,A_n)}{\partial A_{n-1}}$ is ≤ 0 and convex (i.e. $\frac{\partial^3 TC(B_n,A_n)}{\partial A_{n-1}^3} \geq 0$), $\frac{\partial TC(B_n,A_n)}{\partial A_{n-1}} \leq 0$ is an earlier result for $v \gtreqless 0$. Now we look at $\frac{\partial^3 TC(B_n,A_n)}{\partial A_{n-1}^3}$:

$$\frac{\partial^3 TC(B_n, A_n)}{\partial A_{n-1}^3} = \left(\frac{\partial T_n}{\partial A_{n-1}}\right)^2 \frac{\partial^3 TC(B_n, A_n)}{\partial T_{n-1}^3} + \frac{\partial^2 TC(B_n, A_n)}{\partial T_{n-1}^2}\frac{\partial^2 T_n}{\partial A_{n-1}^2},$$

$$\frac{\partial^2 T_n}{\partial A_{n-1}^2} = \frac{2v^2(T_n+1)^2}{u+vM}$$

After taking the derivatives and arranging terms, we obtain the following:

$$\frac{\partial^3 TC(B_n, A_n)}{\partial A_{n-1}^3} = cv\left(\frac{4G(T_n)^3 - 3\sqrt{2}G(T_n)^2 + \sqrt{2}}{G(T_n)^3}\right)$$

and the numerator reduces to $4\sqrt{T_n^2 + 2T_n + 2} - 3\sqrt{2}$, which is ≥ 0 for $T_n \geq 0 (v \geq 0)$. Thus $\frac{\partial TC(B_n,A_n)}{\partial A_{n-1}}$ is a convex function of $A_{n-1}$for $v \geq 0$.

(ii) When $v \leq 0$, $\frac{\partial TC(B_n,A_n)}{\partial A_{n-1}}$ is ≤ 0 and concave (i.e. $\frac{\partial^3 TC(B_n,A_n)}{\partial A_{n-1}^3} \leq 0$),

In the numerator of $\frac{\partial^3 TC(B_n,A_n)}{\partial A_{n-1}^3}$ we have

$$\left(4G(T_n) - 3\sqrt{2}\right) G(T_n)^2 + \sqrt{2}$$

which attains its minimum at $T_n = -1$ in the domain of $T_n \in [-1, \infty]$, since $G(T_n)$ is a strictly increasing function between $[-1, \infty]$. Further,

$$\left(4G(T_n) - 3\sqrt{2}\right) G(T_n)^2 + \sqrt{2} \geq 0$$

and since there is v in the expression, $\frac{\partial^3 TC(B_n,A_n)}{\partial A_{n-1}^3} \leq 0$. Thus $\frac{\partial TC(B_n,A_n)}{\partial A_{n-1}}$ is a concave function of A_{n-1} for $v \leq 0$.

c) Lastly, we show that:

$$\frac{\partial TC(B_{n-1}, A_{n-1})}{\partial A_{n-1}} + \frac{\partial TC(B_n, A_n)}{\partial A_{n-1}} \leq 0 \; when \; A_{n-1} = 0$$

At $A_{n-1} = 0$, $T_{n-1} = 0$ and $T_n = \frac{v(M - B_{n-1})}{u + vB_{n-1}} \geq 0$. Further $\frac{\partial TC(B_{n-1},A_{n-1})}{\partial A_{n-1}} = 0$ when $T_{n-1} = 0$, and from **part (b)** we know that $\frac{\partial TC(B_n,A_n)}{\partial A_{n-1}} < 0 (= 0)$ when $T_n > 0 (= 0)$ or equivalently $M > B_{n-1}$ $(= B_{n-1})$. Hence, $\frac{\partial TC(B_{n-1},A_{n-1})}{\partial A_{n-1}} + \frac{\partial TC(B_n,A_n)}{\partial A_{n-1}} < 0 (= 0)$ when $M > B_{n-1}$ $(= B_{n-1})$ at $A_{n-1} = 0$.

To summarize the results obtained in parts (a),(b) and (c):

a) $TC(B_{n-1}, A_{n-1})$ is a convex non-decreasing function of A_{n-1} for any given B_{n-1},

b) $TC(B_n, A_n)$ is a strongly quasi-convex and non-increasing function of A_{n-1} (when $A_n = M - B_{n-1} - A_{n-1}$ and $B_n = B_{n-1} + A_{n-1}$) and $\frac{\partial TC(B_n,A_n)}{\partial A_{n-1}}$

is convex (concave) for $v \geq 0 (\leq 0)$.

c) $\frac{\partial TC(B_{n-1},A_{n-1})}{\partial A_{n-1}} + \frac{\partial TC(B_n,A_n)}{\partial A_{n-1}} \leq 0$ when $A_{n-1} = 0$

In the light of these results, we conclude that $TC_{n-1,2}(B_{n-1}, A_{n-1}) = \{TC(B_{n-1}, A_{n-1}) + TC(B_n, A_n),\ s.t.\ B_{j+1} = B_j + A_j \text{ for } j = n-1, n\}$ is a strongly quasi-convex(unimodal) function of A_{n-1}.

STEP 2.

In the second step of our proof, we prove that $TC_{n-2,3}(B_{n-2}, A_{n-2}) = \{TC(B_{n-2}, A_{n-2}) + TC(B_{n-1}, A_{n-1}) + TC(B_n, A_n),\ s.t.\ B_{j+1} = B_j + A_j$ for $j = n-2, n-1, n\}$ is a strongly quasi-convex function of A_{n-2} for any given B_{n-2} at a solution A^*_{n-1} satisfying necessary conditions (19),(20) and (21).

Note that $TC_{n-2,3}(B_{n-2}, A_{n-2})$ is a univariate function of A_{n-2} in which B_{n-2} is a parameter and A^*_{n-1} satisfies the corresponding first order optimality conditions. This proof is along the lines of **STEP 1**, and we here prove the following:

a) $TC(B_{n-2}, A_{n-2})$ **is a convex non-decreasing function of** A_{n-2},

b) $TC(B_{n-1}, A^*_{n-1}) + TC(B_n, A^*_n)$ **is a strongly quasi-convex non-increasing function of** A_{n-2}. **Further** $\frac{\partial\left(TC(B_{n-1},A^*_{n-1})+TC(B_n,A^*_n)\right)}{\partial A_{n-2}}$ **is a convex(concave) function of** A_{n-2} **when** $v \geq 0$ (≤ 0).

c) $\frac{\partial TC(B_{n-2},A_{n-2})}{\partial A_{n-2}} + \frac{\partial\left(TC(B_{n-1},A^*_{n-1})+TC(B_n,A^*_n)\right)}{\partial A_{n-2}} > 0$ **when** $A_{n-2} = 0$.

a) $TC(B_{n-2}, A_{n-2})$ is non-decreasing $\left(\text{i.e. } \frac{\partial TC(B_{n-2},A_{n-2})}{\partial A_{n-2}} \geq 0\right)$ and convex in A_{n-2} can be showed in the same way as in previous result of **(a)** at **STEP 1**.

b) If we can show that $\frac{\partial\left(TC(B_{n-1},A^*_{n-1})+TC(B_n,A^*_n)\right)}{\partial A_{n-2}} \leq 0$ for $v \gtreqless 0$, then

$TC(B_{n-1}, A^*_{n-1}) + TC(B_n, A^*_n)$ is a strongly quasi-convex non-increasing function of A_{n-2}.

Observe that:

$$\frac{\partial\left(TC(B_{n-1},A^*_{n-1})+TC(B_n,A^*_n)\right)}{\partial A_{n-2}}=\frac{\partial\left(TC(B_{n-1},A^*_{n-1})+TC(B_n,A^*_n)\right)}{\partial B_{n-1}}\frac{\partial B_{n-1}}{\partial A_{n-2}}$$

$$\underset{\frac{\partial B_{n-1}}{\partial A_{n-2}}=1}{=}\frac{\partial TC(B_{n-1},A^*_{n-1})}{\partial T_{n-1}}\frac{\partial T^*_{n-1}}{\partial B_{n-1}}+\frac{\partial TC(B_n,A^*_n)}{\partial T_n}\frac{\partial T^*_n}{\partial B_{n-1}}$$

$\frac{\partial TC(B_{n-1},A^*_{n-1})}{\partial T_{n-1}}\geq 0\ (\leq 0)$ and $\frac{\partial TC(B_n,A^*_n)}{\partial T_n}\geq 0\ (\leq 0)$ for $v\geq 0\ (\leq 0)$ is already shown in **STEP 1**. The following can be shown from first-order necessary conditions of $TC_{n-1,2}(B_{n-1},A^*_{n-1})$:

$$T^*_{n-1}=\frac{-4D_1+\sqrt{4D_1^2+4D_2^2+2\sqrt{2D_1^4+2D_2^4+12D_1^2D_2^2}}}{D_1}$$

where $D_1=u+vB_{n-1}$ and $D_2=u+vM$.

$$T^*_n=\frac{D_2}{D_1(T^*_{n-1}+1)}-1$$

$$\frac{\partial T^*_{n-1}}{\partial B_{n-1}}=\frac{\partial T^*_{n-1}}{\partial D_1}\frac{\partial D_1}{\partial B_{n-1}}=-\psi_{n-1}(D_1,D_2)v$$

$$\frac{\partial T^*_n}{\partial B_{n-1}}=\frac{\partial T^*_n}{\partial D_1}\frac{\partial D_1}{\partial B_{n-1}}=-\psi_n(D_1,D_2)v$$

where $\psi_{n-1}$and ψ_n are positive valued functions for $D_1,D_2\geq 0$ and $v\gtreqqless 0$.

Therefore, $\frac{\partial T^*_{n-1}}{\partial B_{n-1}}\leq 0\ (\geq 0)$ and $\frac{\partial T^*_n}{\partial B_{n-1}}\leq 0(\geq 0)$ for $v\geq 0\ (\leq 0)$. Thus $\frac{\partial\left(TC(B_{n-1},A^*_{n-1})+TC(B_n,A^*_n)\right)}{\partial A_{n-2}}\leq 0$ and $TC(B_{n-1},A^*_{n-1})+TC(B_n,A^*_n)$ is a non-increasing and quasi-convex function of A_{n-2}.

Further, we need to show that $\frac{\partial\left(TC(B_{n-1},A^*_{n-1})+TC(B_n,A^*_n)\right)}{\partial A_{n-2}}$ is a convex(concave) function of A_{n-2} when $v\geq 0\ (\leq 0)$. This proof is again along the lines of previous results and could be shown from the second order derivative of $\frac{\partial^3\left(TC(B_{n-1},A^*_{n-1})+TC(B_n,A^*_n)\right)}{\partial A^3_{n-2}}$ using the following relation.

$$\frac{\partial^3\left(TC(B_{n-1},A^*_{n-1})+TC(B_n,A^*_n)\right)}{\partial A^3_{n-2}} \underset{\frac{\partial B_{n-1}}{\partial A_{n-2}}=1}{=} \frac{\partial^3\left(TC(B_{n-1},A^*_{n-1})+TC(B_n,A^*_n)\right)}{\partial B^3_{n-1}}$$

$$\begin{aligned} = \; & \left(\frac{\partial T^*_{n-1}}{\partial B_{n-1}}\right)^2\left(\frac{\partial^3 TC(B_{n-1},A^*_{n-1})}{\partial T^3_{n-1}}\right)+\left(\frac{\partial T^*_n}{\partial B_{n-1}}\right)^2\left(\frac{\partial^3 TC(B_n,A^*_n)}{\partial T^3_n}\right) \\ & +\frac{\partial^2 T^*_{n-1}}{\partial B^2_{n-1}}\left(\frac{\partial^2\left(TC(B_{n-1},A^*_{n-1})+TC(B_n,A^*_n)\right)}{\partial T^2_{n-1}}\right)\end{aligned}$$

In the above equation, last term cancels out since (A^*_{n-1},A^*_n) satisfies the first-order condition $\frac{\partial\left(TC(B_{n-1},A^*_{n-1})+TC(B_n,A^*_n)\right)}{\partial T_{n-1}}=0$. In the two remaining terms, squares are obviously positive for $v \gtreqless 0$, and from previous results we obtain that

$\frac{\partial\left(TC(B_{n-1},A^*_{n-1})+TC(B_n,A^*_n)\right)}{\partial A_{n-2}}$ is a convex(concave) function of A_{n-2} when $v \geq 0$ (≤ 0) .

c) Lastly we show that:

$$\frac{\partial TC(B_{n-2},A_{n-2})}{\partial A_{n-2}}+\frac{\partial\left(TC(B_{n-1},A^*_{n-1})+TC(B_n,A^*_n)\right)}{\partial A_{n-2}}<0(=0) \text{ when } A_{n-2}=0.$$

We know that at $A_{n-2}=0$, $T_{n-2}=0$ thus $\frac{\partial TC(B_{n-2},A_{n-2})}{\partial A_{n-2}}=0$.

$$\begin{aligned}\frac{\partial\left(TC(B_{n-1},A^*_{n-1})+TC(B_n,A^*_n)\right)}{\partial A_{n-2}} = \; & \frac{\partial TC(B_{n-1},A^*_{n-1})}{\partial T_{n-1}}\frac{\partial T^*_{n-1}}{\partial B_{n-1}} \\ & +\frac{\partial TC(B_n,A^*_n)}{\partial T_n}\frac{\partial T^*_n}{\partial B_{n-1}}\end{aligned}$$

As shown in **part (b)**, $\frac{\partial\left(TC(B_{n-1},A^*_{n-1})+TC(B_n,A^*_n)\right)}{\partial A_{n-2}} \leq 0$ for $v \gtreqless 0$ and at $A_{n-2}=$

0 the equality only holds when $T^*_{n-1} = 0$ (i.e. $A^*_{n-1} = 0$) and $T^*_n = 0$ (i.e. $A^*_n = 0$), in other words $B_{n-2} = M$. Since this is a contradiction the existence of the problem, $\frac{\partial\left(TC(B_{n-1},A^*_{n-1})+TC(B_n,A^*_n)\right)}{\partial A_{n-2}} < 0$ for well-defined problems.

In summary, we have first proved in **STEP 1** that $TC_{n-1,2}(B_{n-1}, A_{n-1})$ is a strongly quasi-convex function of A_{n-1} for any given B_{n-1}. Then in **STEP 2**, we have shown that, at any solution A^*_{n-1} satisfying necessary conditions, $TC_{n-2,3}(B_{n-2}, A_{n-2})$ is a strongly quasi-convex function of B_{n-1} and thus unimodal in A_{n-1} and B_{n-1}. **STEP 1** proves that $TC_{n-2,3}(B_{n-2}, A_{n-2})$ is unimodal(convex) in A_{n-1} for any given B_{n-1} or equally for any A_{n-2} given B_{n-2}. **STEP 2** proves that $TC_{n-2,3}(B_{n-2}, A_{n-2})$ is also unimodal in A_{n-2} at the minimizing values of A_{n-1}. Thus total cost in this last triplet of areas is jointly unimodal (strongly quasi-convex) in A_{n-1} and A_{n-2}. As a generalization for any given two boundaries, B_i and B_{i+3}, total cost is jointly unimodal in A_i and A_{i+1}. ■

Proposition 3.3

In the optimal solution, facility locations in every neighboring service region pair are equidistanced from the shared boundary.

Proof.

Rewriting the Euler equation in (26) with $\mathrm{A}_{i+1} = B_{i+2}$-$B_i$$-$$\mathrm{A}_i$ and $B_{i+1} =$

B_i+A_i gives us the below expression.

$$\begin{aligned} & \frac{\partial TC(B_i, A_i)}{\partial A_i} + \frac{dTC(B_{i+1}, A_{i+1})}{dA_i} \\ = & \frac{\partial\left(TC(B_i, A_i)\right)}{\partial A_i} + \frac{\partial\left(TC(B_i + \mathrm{A}_i, B_{i+2} - B_i - \mathrm{A}_i)\right)}{\partial A_i} \\ = & \frac{\partial\left(TC(B_i, A_i) + TC(B_i + \mathrm{A}_i, B_{i+2} - B_i - \mathrm{A}_i)\right)}{\partial A_i} = 0 \end{aligned} \tag{95}$$

Recalling that total cost for each service region is calculated as follows:

$$\begin{aligned} TC(B_i, A_i) &= F + f + a\mathrm{A}_i D(\left(B_i + \frac{\mathrm{A}_i}{2}\right) \\ & \quad + cK(\left(\frac{v\mathrm{A}_i}{u + vB_i}\right) \mathrm{A}_i^2 D(\left(B_i + \frac{\mathrm{A}_i}{2}\right) \\ TC(B_i + \mathrm{A}_i, B_{i+2} - B_i - \mathrm{A}_i) &= F + f \\ & \quad + a(B_{i+2} - B_i - \mathrm{A}_i) D(B_i + \mathrm{A}_i + \frac{B_{i+2} - B_i - \mathrm{A}_i}{2}) \end{aligned}$$

$$+cK\left(\frac{v(B_{i+2} - B_i - \mathrm{A}_i)}{u + v(B_i + \mathrm{A}_i)}\right)(B_{i+2} - B_i - \mathrm{A}_i)^2 D\left(B_i + \mathrm{A}_i + \frac{B_{i+2} - B_i - \mathrm{A}_i}{2}\right)$$

We can now explicitly express total cost for service regions $\mathcal{A}_i$ and $\mathcal{A}_{i+1}$ accordingly.

$$\begin{aligned} & TC(B_i, A_i) + TC(B_i + \mathrm{A}_i, B_{i+2} - B_i - \mathrm{A}_i) \\ = & \ 2(F + f) + a\left(u(B_{i+1} + B_i) + \frac{v(B_{i+2}^2 - B_i^2)}{2}\right) \\ & + cK\left(\frac{v\mathrm{A}_i}{u + vB_i}\right) \mathrm{A}_i^2 D\left(B_i + \frac{\mathrm{A}_i}{2}\right) \\ & + cK\left(\frac{v(B_{i+2} - B_i - \mathrm{A}_i)}{u + v(B_i + \mathrm{A}_i)}\right)(B_{i+2} - B_i - \mathrm{A}_i)^2 D\left(B_i + \mathrm{A}_i + \frac{B_{i+2} - B_i - \mathrm{A}_i}{2}\right) \end{aligned} \tag{96}$$

Note that in (96), only the last two terms, transportation costs in $\mathcal{A}_i$ and $\mathcal{A}_{i+1}$, have explicit dependence on A_i. Hence remaining terms of (96) cancel out in the differentiation operation. The elevation factors for $\mathcal{A}_i$ and $\mathcal{A}_{i+1}$ are defined as follows:

$$T_i = \frac{v\mathrm{A}_i}{u+vB_i} \tag{97}$$

$$T_{i+1} = \frac{v(\mathrm{A}_{i+1})}{u+v(B_{i+1})} = \frac{v(B_{i+2}-B_i-\mathrm{A}_i)}{u+v(B_i+\mathrm{A}_i)} \tag{98}$$

Using the result in (11) we could express transportation costs, the last two terms of (96) in terms of (97) and (98) as below.

$$\text{Transportation Cost in } \mathcal{A}_i = \frac{(4-\frac{Z(T_i)^3}{2}+2T_i^3+6T_i^2+6T_i)(u+vB_i)^3}{6v^2}$$

$$\begin{aligned}&\text{Transportation Cost in } \mathcal{A}_{i+1}\\ =\ &\frac{(4-\frac{Z(T_{i+1})^3}{2}+2T_{i+1}^3+6T_{i+1}^2+6T_{i+1})(u+v(B_i+A_i))^3}{6v^2}\end{aligned}$$

$$Z(T)=\sqrt{2T^2+4T+4}$$

Expression (95) becomes:

$$\frac{\partial(\text{Transportation Cost in } \mathcal{A}_i + \text{Transportation Cost in } \mathcal{A}_{i+1})}{\partial A_i} \tag{99}$$

$$= \frac{\partial\left(\frac{(4-\frac{Z(T_i)^3}{2}+2T_i^3+6T_i^2+6T_i)(u+vB_i)^3}{6v^2}+\frac{(4-\frac{Z(T_{i+1})^3}{2}+2T_{i+1}^3+6T_{i+1}^2+6T_{i+1})(u+v(B_i+A_i))^3}{6v^2}\right)}{\partial A_i} = 0 \tag{100}$$

For convenience, first order condition in (99) can be revised for partial differentiation with respect to the elevation factor T_i if we observe the following equivalence.

$$\frac{\partial(\cdot)}{\partial A_i} = \frac{v}{(u+vB_i)}\frac{\partial(\cdot)}{\partial T_i} \tag{101}$$

Note that in (101) B_i is subjec to to $B_i \leq M$, so equality holds true only when the differentiation is equal to zero. For simplicity in disposition, it is better to evaluate the differentiation in (99) separately for the two transportation terms.

$$\frac{\partial(\text{Transportation Cost in } \mathcal{A}_i)}{\partial T_i} = \frac{\left(-3\frac{Z(T_i)^2}{2}\frac{\partial Z(T_i)}{\partial T_i} + 6T_i^2 + 12T_i + 6\right)(u+vB_i)^3}{6v^2} \tag{102}$$

$$\begin{aligned} &\frac{\partial(\text{Transportation Cost in } \mathcal{A}_{i+1})}{\partial T_i} = \\ = &\frac{\left(-3\frac{Z(T_{i+1})^2}{2}\frac{\partial Z(T_{i+1})}{\partial T_{i+1}}\frac{\partial T_{i+1}}{\partial T_i} + (6T_i^2 + 12T_i + 6)\frac{\partial T_{i+1}}{\partial T_i}\right)(u+v(B_i+A_i))^3}{6v^2} \\ &+\frac{(4 - \frac{Z(T_{i+1})^3}{2} + 2T_{i+1}^3 + 6T_{i+1}^2 + 6T_{i+1})(1+T_i)^2(u+vB_i)^3}{6v^2} \end{aligned} \tag{103}$$

Note that

$$u + v(B_i + A_i) = (u+vB_i)(1+T_i) \tag{104}$$

$$(1+T_i)(1+T_{i+1}) = \frac{(u+vB_{i+2})}{(u+vB_i)} \tag{105}$$

$$\frac{\partial Z(T_i)}{\partial T_i} = \frac{2(T_i+1)}{Z(T_i)} \tag{106}$$

$$\frac{\partial T_{i+1}}{\partial T_i} = \frac{-(u+vB_{i+2})}{(T_i+1)^2(u+vB_i)} \tag{107}$$

After some algebraic manipulations by using (106) and (107), partial derivatives in (102) and (103) results in the following expressions.

$$\frac{\partial(\text{Transportation Cost in } \mathcal{A}_i)}{\partial T_i} = \frac{(-Z(T_i) + 2T_i + 2)(T_i + 1)(u + vB_i)^3}{2v^2}$$

$$\begin{aligned} &\frac{\partial(\text{Transportation Cost in } \mathcal{A}_{i+1})}{\partial T_i} = \\ = \;& (u + vB_{i+2})(T_i + 1)(u + vB_i)^2 \frac{(T_{i+1}Z(T_{i+1}) + Z(T_{i+1}) - Z(T_{i+1})^2 + 2)}{2v^2} \\ &+ \frac{(4 - \frac{Z(T_{i+1})^3}{2} + 2T_{i+1}^3 + 6T_{i+1}^2 + 6T_{i+1})(1 + T_i)^2(u + vB_i)^3}{6v^2} \end{aligned}$$

Through variable substitution using (105) and additional manipulations, expression (99) transforms to the following expression.

$$\begin{aligned} &(Z(T_i) - 2T_i - 2)(u + vB_i) - \frac{(u + vB_{i+2})}{(T_{i+1} + 1)} \\ &\left[(T_{i+1} + 1)^2(Z(T_{i+1}) - 2T_{i+1} - 2) + 4 - \frac{Z(T_{i+1})^3}{2} + 2T_{i+1}^3 + 6T_{i+1}^2 + 6T_{i+1}\right] = 0 \end{aligned}$$

Above expression further reduces to,

$$\left(1 - \frac{(Z(T_i) - 2)}{2T_i}\right) A_i = (B_{i+2} - B_i - A_i)\left(\frac{Z(T_{i+1}) - 2}{2T_{i+1}}\right) \tag{108}$$

From the earlier result in (97), one can derive the following expression:

$$\frac{(Z(T_i) - 2)}{2T_i} A_i = x_i - B_i \tag{109}$$

When (109) is substituted in (108), we obtain the following:

$$A_i - x_i + B_i = x_{i+1} - B_{i+1}$$

which is the property in (27). ■

9.2 Appendix 4

9.2.1 Improvement Based- Sequential Location-Allocation (SLA) Method

Second solution approach in Section 4.4. is the sequential location-allocation (SLA) method. Eventhough this method has been introduced in the literature long time ago (Cooper 1964) for location-allocation problems with discrete demand, we extend this approach to case where the demand is in the form of a continuous function.

Sequential Location-Allocation algorithm (SLA), due to (Cooper, 1964), is an inexact first-order solution procedure which alternates between location and allocation variable spaces. Originally, this procedure has been suggested for planar location-allocation problems with discrete demand data. In this section, we will extend this approach to planar location-allocation problems with **continuous demand data**. Although this extension is merely an algorithmic adaptation of SLA method for continuous demand, it nevertheless represents a missing component of SLA method and constitutes the linkage between location-allocation problem in allocation (LAM-AVS) and location variable spaces (LAM-LVS).

We first present the formal algorithm for the SLA method for continuous demand functions and then provide an example for the case of Euclidean-metric (L_2).

Sequential Location-Allocation (SLA) Method:

Step 1. Define and Initialize the model parameters and variables

j : index for the feasibility iterations

ϵ_{BOUND} : epsilon parameter for optimality stopping decision

(i.e. optimality stopping iteration $j^* = \{j| \ \epsilon_{BOUND} \geq \frac{|TC^j - TC^{j-1}|}{TC^{j-1}}\}$

Set $j = 1$, and select a feasible solution:

- Allocation decisions $(\mathcal{A}_1)^j$ and $(\mathcal{A}_2)^j$
- Location decisions $(\mathbf{x}_1)^j$ and $(\mathbf{x}_2)^j$

Step 2. Solve Problem in Location Variable Space (LAM-LVS) by relaxing constraint (41)

Solve Single-facility problem in $(\mathcal{A}_1)^j$ and $(\mathcal{A}_2)^j$

Assign solution to $(\mathbf{x}_1^*)^j$ and $(\mathbf{x}_2^*)^j$

Step 3. Solve Problem in Allocation Variable Space (LAM-AVS) by relaxing constraint (43)

Find Nearest-Neighbor solution for $(\mathbf{x}_1^*)^j$ and $(\mathbf{x}_2^*)^j$

Assign solution to $(\mathcal{A}_1)^j$ and $(\mathcal{A}_2)^j$

Step 4. Check for optimality

Stop if $\epsilon_{BOUND} \geq \frac{|TC^j - TC^{j-1}|}{TC^{j-1}}$ else assign for i=1,2

$(\mathbf{x}_i^*)^{j+1} \leftarrow (\mathbf{x}_i^*)^j$

$(\mathcal{A}_i)^{j+1} \leftarrow (\mathcal{A}_i)^j$

Return to **Step 2.**

Let's illustrate the SLA algorithm with an example based on Euclidean-metric (L_2). For consistency and comparison purposes, we choose the same example as the one previously presented. Our market region is a square , $\mathcal{M}$=$\{(x, y)|x \in (0, 100)$, and $y \in (0, 100)\}$, i.e. $M = 100$ and demand density function is linear ($D(x, y) = 100 + 10x + 5y$) over the market region $\mathcal{M}$. The

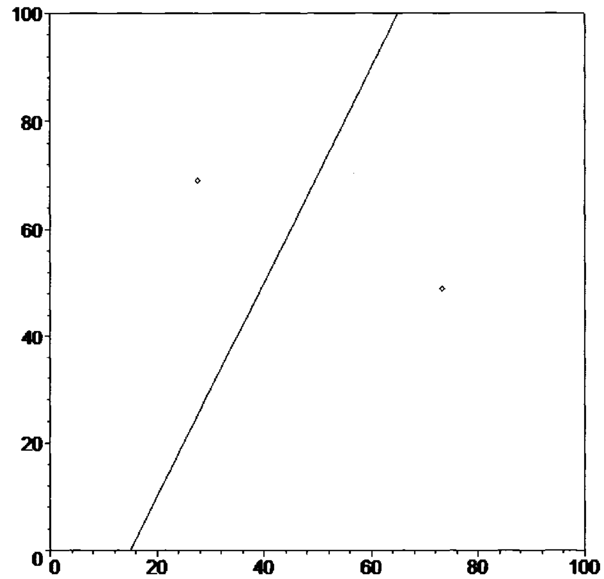

Figure 9.25: Starting solution of the Euclidean-metric based example for Sequential Location-Allocation Method.

starting solution is illustrated in Figure 9.25.

This starting solution, both the locations and the allocation decisions, correspond to the example presented in the constructive solution method section where initial triggers were $A1$=35 and $A2$ =40 at $y_{P1} = A3 = 40$ and $y_{P2} = A4 = 50$, respectively.

The steps of the algorithm is as follows:

ITERATION 1

Step 1. Define and Initialize the model parameters and variables

$\epsilon_{BOUND} = 1.0 \times 10^{-4}$

Allocation Decisions:

Allocation Line BR

Slope $a := 2$, Intercept $b := -30$

$br(x) = y = 2x - 30$

Service Regions

$$(\mathcal{A}_1)^{j=0} := \{(x,y)|y \in Y_{BR} = [0,100] \text{ and } x \in [0, \tfrac{y}{2} + 15]\}$$

$$(\mathcal{A}_2)^{j=0} := \{(x,y)|y \in Y_{BR} = [0,100] \text{ and } x \in [\tfrac{y}{2} + 15, 100]\}$$

Location Decisions:

$$(\mathbf{x}_1)^{j=0} = (x_1, y_1) = (27.720, 69.071)$$

$$(\mathbf{x}_2)^{j=0} = (x_2, y_2) = (73.267, 48.956)$$

Total Cost:

$$TC^{j=0}(\mathcal{A}_1, \mathcal{A}_2, \mathbf{x}_1, \mathbf{x}_2) = 248,433,712.40$$

Step 2. Solve Problem in Location Variable Space (LAM-LVS) by relaxing constraint (41)

Solve Single-facility median problem using Weiszfeld's Method

$$(\mathbf{x}_1^*)^{j=1} = (x_1^*, y_1^*) = \arg\min_{(\mathbf{x}_1)} \int_{(\mathcal{A}_1)^{j=0}} d_{L_2}(\mathbf{x}_1, \mathbf{x}) D(\mathbf{x}) d\mathbf{x} = (28.742, 71.890)$$

$$(\mathbf{x}_2^*)^{j=1} = (x_2^*, y_2^*) = \arg\min_{(\mathbf{x}_2)} \int_{(\mathcal{A}_2)^{j=0}} d_{L_2}(\mathbf{x}_2, \mathbf{x}) D(\mathbf{x}) d\mathbf{x} = (74.213, 49.946)$$

Step 3. Solve Problem in Allocation Variable Space (LAM-AVS) by relaxing constraint (43)

Solve following for Nearest-Neighbor solution:

$$d_{L_2}((\mathbf{x}_1)^{j=1}, (A^x(y), A^y(x))) - d_{L_2}((\mathbf{x}_2)^{j=1}, (A^x(y), A^y(x))) = 0$$

Allocation Decisions:

Allocation Line BR

Slope $a := 2.072$, Intercept $b := -45.750$

$br(x) = y = 2.072x - 45.750,$

Service Regions

$(\mathcal{A}_1)^{j=1} := \{(x,y)|y \in Y_{BR} = [0,100] \text{ and } x \in [0, 0.4826y +$

22.0790]}

$$(\mathcal{A}_2)^{j=1} := \{(x,y)|y \in Y_{BR} = [0,100] \text{ and } x \in [0.4826y+22.0790, 100]\}$$

Step 4. Check for optimality

Total Cost:

$$TC^{j=1}(\mathcal{A}_1, \mathcal{A}_2, \mathbf{x}_1^*, \mathbf{x}_2^*) = 246,204,796.30$$

$$\frac{|TC^{j=1} - TC^{j=0}|}{TC^{j=0}} = 9.0 \times 10^{-3} > \epsilon_{BOUND}$$

Figure 9.26 illustrates the solution at the end of first iteration.

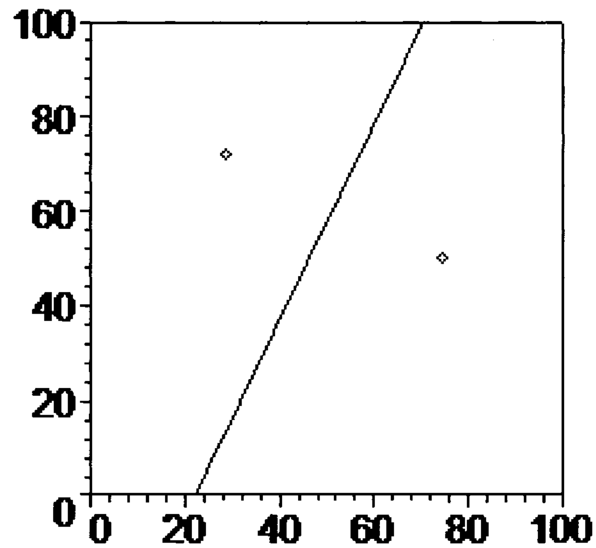

Figure 9.26: Solution at the end of first iteration of the Euclidean-metric based example for Sequential Location-Allocation Method.

For brevity, illustration of a single iteration would suffice to understand the SLA approach. When continued with the SLA method, by the end of **32nd iteration,** solution displayed in Figure 9.27 is obtained. Iteration steps are summarized in Table A.4.1. Note that, despite the three-fold iteration

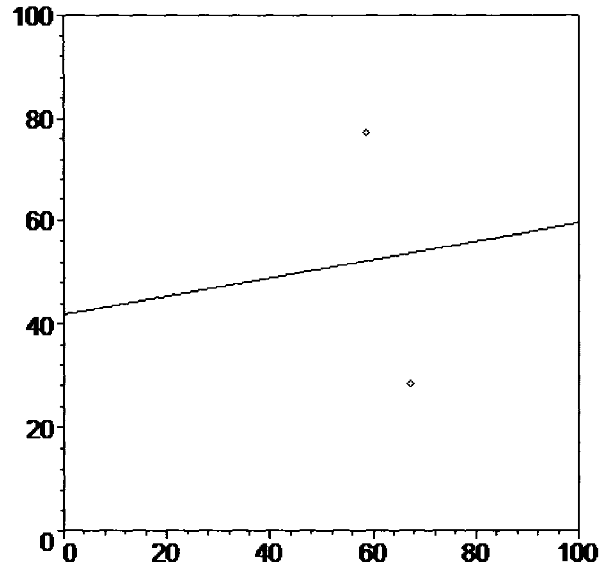

Figure 9.27: Solution at the end of 32^{nd} iteration of the Euclidean-metric based example for Sequential Location-Allocation Method.

count, final objective function value is still worse than the one obtained in the constructive approach.

Iteration(j)	Variable Space	Type	x1	y1	x2	y2	Slope	Intercept	TC
0		I	27.72	69.07	73.27	48.96	2.000	-30.000	248,433,712.40
1	Location	I	28.74	71.89	74.21	49.95	2.000	-30.000	248,018,074.90
	Allocation	I	28.74	71.89	74.21	49.95	2.072	-45.750	246,204,796.30
2	Location	I	31.97	70.10	76.31	48.85	2.072	-45.750	245,199,184.10
	Allocation	I	31.97	70.10	76.31	48.85	2.087	-53.509	244,603,093.00
3	Location	I	33.76	69.24	77.39	48.02	2.087	-53.509	244,286,638.40
	Allocation	I	33.76	69.24	77.39	48.02	2.056	-55.638	244,099,369.20
4	Location	I	34.85	68.94	77.96	47.44	2.056	-55.638	243,988,299.20
	Allocation	I	34.85	68.94	77.96	47.44	2.005	-54.913	243,920,560.50
5	Location	I	35.65	69.06	78.28	46.99	2.005	-54.913	243,869,818.80
	Allocation	I	35.65	69.06	78.28	46.99	1.931	-51.998	243,833,019.00
6	Location	I	36.17	69.21	78.36	46.49	1.931	-51.998	243,803,494.30
	Allocation	I	36.17	69.21	78.36	46.49	1.857	-48.467	243,779,915.00
...	...		...	...	...	...	...	...	...
21	Location	IV	45.41	75.01	75.49	37.46	0.974	-1.625	242,236,124.40
	Allocation	IV	45.41	75.01	75.49	37.46	0.801	7.809705	241,753,555.30
22	Location	III	48.21	76.08	74.38	35.16	0.801	7.809705	240,295,442.80
	Allocation	III	48.21	76.08	74.38	35.16	0.64	16.41292	239,541,760.20
25	Location	III	56.63	77.28	69.14	29.54	0.316	33.821	236,592,568.70
	Allocation	III	56.63	77.28	69.14	29.54	0.262	36.924	236,491,785.10
29	Location	III	59.02	77.49	67.24	28.66	0.198	40.873	236,386,847.00
	Allocation	III	59.02	77.49	67.24	28.66	0.168	42.451	236,352,293.60
32	Location	III	58.67	77.32	67.27	28.50	0.173	42.067	236,346,792.00
	Allocation	III	58.67	77.32	67.27	28.50	0.176	41.816	236,345,501.30

Table A.1: Sequential Location-Allocation Algorithm's iteration results based on Euclidean-metric (Example A.1).

Proposition 4.1.

The optimal locations of the two facilities ($\mathbf{x}_1^*$ and $\mathbf{x}_2^*$)*, given the allocation decisions, satisfy the following conditions when the distance measure is based on the* **Squared Euclidean – Metric** $(\mathbf{L}_2^2)$:

$$(x_i^*, y_i^*) = \left(x_i^G, y_i^G\right) \qquad \text{for } i = 1, 2$$

where x_i^G *and* y_i^G *are the* $x-$ *and* $y-$ *dimensional centroids of* $A_{i=1,2}$ *with respect to* $D(x)$.

$$x_i^G = \frac{\int\limits_{\mathcal{A}_i} xD(\mathbf{x})d\mathbf{x}}{\int\limits_{\mathcal{A}_i} D(\mathbf{x})d\mathbf{x}} \qquad \text{and} \qquad y_i^G = \frac{\int\limits_{\mathcal{A}_i} yD(\mathbf{x})d\mathbf{x}}{\int\limits_{\mathcal{A}_i} D(\mathbf{x})d\mathbf{x}} \qquad \text{for } i = 1, 2$$

Proof.

Since the proof is identical for x and y dimensions, we will only prove for x_i. For a given allocation solution $(\mathcal{A}_i, i = 1, 2)$, define the objective function as $TC(\mathcal{A}_i, i = 1, 2)$. When we take the partial derivative of the $TC(\mathcal{A}_i, i = 1, 2)$ with respect to x_i, the only differential term is due to the transportation cost component of $TC(\mathcal{A}_i, i = 1, 2)$, which is $\int\limits_{\mathcal{A}_i} [(x_i - x)^2 + (y_i - y)^2] D(\mathbf{x})d\mathbf{x}$.Hence, we have the following steps.

$$\begin{aligned}
\frac{\partial TC(\mathcal{A}_i, i = 1, 2)}{\partial x_i} &= 2\int\limits_{y}\int\limits_{\mathcal{A}_i^x(y)} (x_i - x)D(\mathbf{x})dxdy \\
&= 2x_i\int\limits_{y}\int\limits_{\mathcal{A}_i^x(y)} D(\mathbf{x})dxdy - 2\int\limits_{y}\int\limits_{\mathcal{A}_i^x(y)} xD(\mathbf{x})dxdy \qquad (110) \\
&= 2(x_i - x_i^G)W_i \qquad (111)
\end{aligned}$$

where $W_i = \int\limits_{y}\int\limits_{\mathcal{A}_i^x(y)} D(\mathbf{x})dxdy$ is the total demand volume served in service region $\mathcal{A}_i$ and x_i^G, centroid of $\mathcal{A}_i$ with respect to $D(\mathbf{x})$ is defined as follows.

$$x_i^G = \frac{\int\limits_{\mathcal{A}_i} xD(\mathbf{x})d\mathbf{x}}{W_i} = \frac{\int\limits_{\mathcal{A}_i} xD(\mathbf{x})d\mathbf{x}}{\int\limits_{\mathcal{A}_i} D(\mathbf{x})d\mathbf{x}}$$

Thus the optimal location decisions (x_i^*, y_i^*), are the centroids of the designated service regions, $\mathcal{A}_i$ with respect to $D(\mathbf{x})$ for the Squared Euclidian-metric. ∎

Proposition 4.2.

The optimal locations of the two facilities ($\mathbf{x}_1^*$ and $\mathbf{x}_2^*$), *given the allocation* *decisions, satisfy the following conditions when the distance measure is based on the* **Euclidean – Metric** $(\mathbf{L}_2)$:

$$\int_{\mathcal{A}_i} \frac{(x_i^* - x)}{||\mathbf{x}_i^* - \mathbf{x}||} D(\mathbf{x}) d\mathbf{x} = 0 \qquad \text{for } i = 1, 2 \tag{112}$$

$$\int_{\mathcal{A}_i} \frac{(y_i^* - y)}{||\mathbf{x}_i^* - \mathbf{x}||} D(\mathbf{x}) d\mathbf{x} = 0 \qquad \text{for } i = 1, 2 \tag{113}$$

Proof.

Along the lines of proof for Proposition 4.2., we will only prove for x_i. When we take the partial derivative of the $TC(\mathcal{A}_i, i = 1, 2)$ with respect to x_i, the only differential term is due to the transportation cost component of $TC(\mathcal{A}_i, i = 1, 2)$, which is $\int_{\mathcal{A}_i} ||\mathbf{x}_i^* - \mathbf{x}|| \, D(\mathbf{x}) d\mathbf{x}$.Hence, we have the following steps.

$$\frac{\partial TC(\mathcal{A}_i, i = 1, 2)}{\partial x_i} = \int_{\mathcal{A}_i} \frac{(x_i - x)}{||\mathbf{x}_i^* - \mathbf{x}||} D(\mathbf{x}) dx dy = 0 \tag{114}$$

Thus the optimal location decisions (x_i^*, y_i^*) satisfy the relations in (112) and (113) for $\mathcal{A}_i$ with respect to $D(\mathbf{x})$ for Euclidian-metric. ■

Proposition 4.3. *For Euclidean-metric based distance measure cases, the derivative of the single dimensional allocation decisions* ($A^y(x)$ and $A^x(y)$) *with respect to the slope* (a) *of* BR *when rotated around a reference point* (x_r, y_r) *is as follows:*

$$\frac{dA^x(y)}{da_r} = \frac{(y_r - y)}{a^2}$$

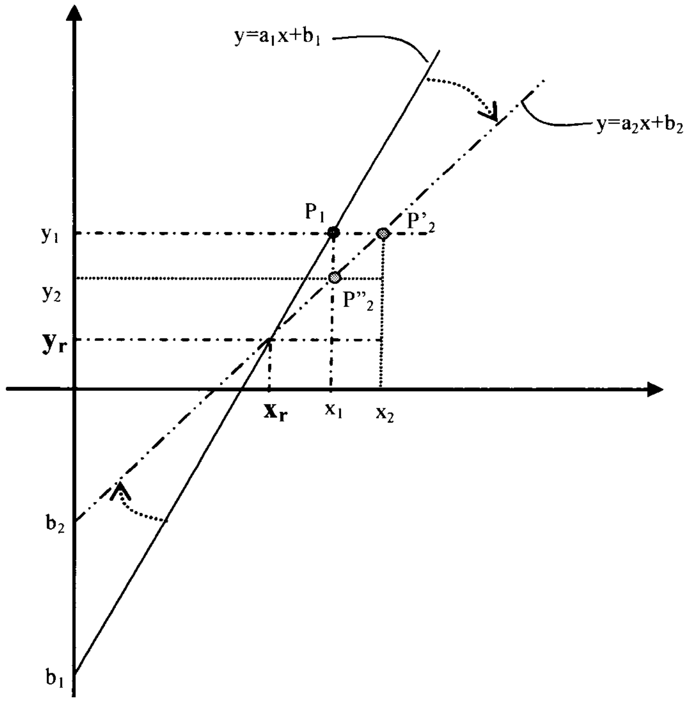

Figure 9.28: Rotating line($y = a_1x + b_1$) around a reference point (x_r, y_r).

$$\frac{dA^y(x)}{da_r} = (x - x_r)$$

Proof.

First, we derive $\left(\frac{\partial A^x(y)}{\partial a}\right)$ using the Figure 9.28. In the figure, (x_r, y_r) is the reference point for our clockwise rotation. As a result of this rotation, equation of the line changes from $y = a_1x + b_1$ to $y = a_2x + b_2$. Accordingly, coordinates of point $P1$, i.e. (x_1, y_1) changes. Hence we could express this change in x_1 as $dA^x(y_1)$, i.e change in single-dimensional allocation decision at level y_1. Note that $y_1 - y_r = a_1(x_1 - x_r)$ and $y_2 - y_r = a_2(x_2 - x_r)$.

$$\begin{aligned} dA^x(y_1) &= x_2 - x_1 \\ &= \frac{y_1 - y_r}{a_2} - \frac{y_1 - y_r}{a_1} \\ &= (y_r - y_1)\left(\frac{1}{a_1} - \frac{1}{a_2}\right) \\ &= (y_r - y_1)\left(\frac{a_2 - a_1}{a_1 a_2}\right) \end{aligned}$$

When we take the limit of the new slope a_2 towards original slope a_1 :

$$\begin{aligned} da_1 &= \lim_{a_2 \to a_1} (a_2 - a_1) \\ \lim_{a_2 \to a_1} a_1 a_2 &= a_1^2 \\ \frac{dA^x(y_1)}{da_1} &= \frac{(y_r - y_1)}{a_1^2} \end{aligned}$$

We can derive$\frac{\partial A^y(x)}{\partial a_1}$ along the lines of above approach.

$$\begin{aligned} dA^y(x_1) &= y_2 - y_1 \\ &= (x_1 - x_r)\, a_2 - (x_1 - x_r)\, a_1 \\ &= (x_1 - x_r)(a_2 - a_1) \\ da_1 &= \lim_{a_2 \to a_1} (a_2 - a_1) \\ \frac{dA^y(x_1)}{da_1} &= (x_1 - x_r) \end{aligned}$$

■

Proposition 4.4. *For Euclidean-metric based distance measure, when we rotate the allocation line BR around a reference point, the following relationship* *holds true irrespective of the reference point.*

$$\frac{dA^y(x)}{da} = (-a)\frac{dA^x(y)}{da} \tag{115}$$
$$\frac{dA^y(x)}{db} = (-a)\frac{dA^x(y)}{db} \tag{116}$$

Proof.

Recall from Proposition 4.3.:

$$\frac{dA^x(y)}{da_r} = \frac{(y_r - y)}{a^2}$$
$$\frac{dA^y(x)}{da_r} = (x - x_r)$$

When we express $(y_r - y) = a(x_r - x)$, below relation proves (115) in the proposition.

$$\frac{dA^x(y)}{da_r} = \frac{(x_r - x)}{a} = \left(\frac{-1}{a}\right)\frac{dA^y(x)}{da_r}$$

In order to prove (116), we first need to derive the results for the translation, i.e.$\frac{dA^y(x)}{db}$ and $\frac{dA^x(y)}{db}$. Figure 9.29 illustrates this transformation.

With the translation (i.e. change in the intercept b), the change in the

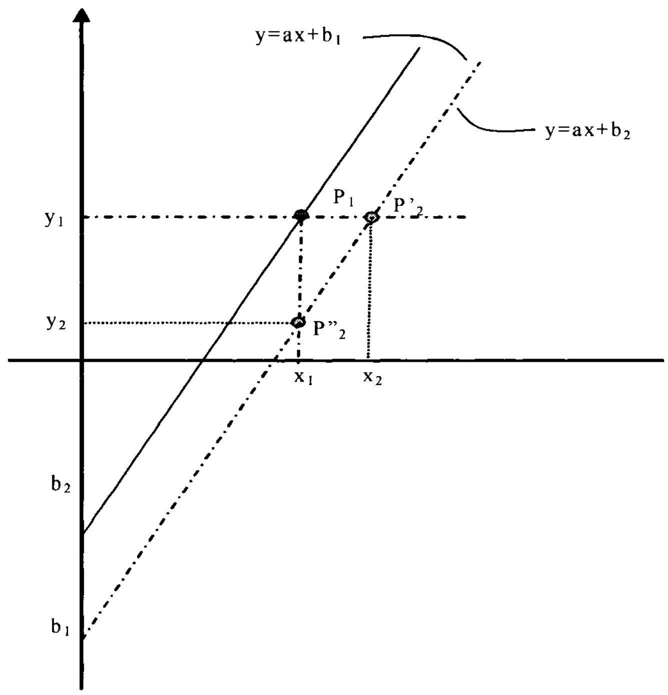

Figure 9.29: Translating line($y = ax + b_1$) by changing the intercept.

horizontal allocation decision $dA^x(y_1)$ is as follows.

$$\begin{aligned}
dA^x(y_1) &= x_2 - x_1 \\
&= \frac{y_1 - b_2}{a} - \frac{y_1 - b_1}{a} \\
&= \frac{b_1 - b_2}{a} = \frac{-(b_2 - b_1)}{a} \\
db &= \lim_{b_2 \to b_1} (b_2 - b_1) \\
\frac{dA^x(y_1)}{db} &= \frac{-1}{a}
\end{aligned}$$

Accordingly, the change in the vertical allocation decision $dA^y(x_1)$ is as follows.

$$\begin{aligned}
dA^y(x_1) &= y_2 - y_1 \\
&= ax_1 + b_2 - (ax_1 + b_1) \\
&= (b_2 - b_1) \\
db &= \lim_{b_2 \to b_1} (b_2 - b_1) \\
\frac{dA^y(x_1)}{db} &= 1
\end{aligned}$$

Note that as in the case of rotation around a reference point, the following relationship holds true for translation as well (for brevity exluded).

$$\frac{dA^y(x)}{db} = (-a)\frac{dA^x(y)}{db}$$

Proposition 4.5. *Rotation around a reference point (x_r, y_r) by (da) is an equivalent transformation to first decreasing the intercept by $x_r(da)$ and then performing a pure rotation around $(0, b)$.*

Proof.

Now we establish the equivalence of rotation around a reference point (x_r, y_r) with the translation and pure rotation. Figure 9.30 illustrates transforming line $(y = a_1x + b_1)$ by first translation and then rotation.

From the above illustration, the value $\Delta = b_2 - b_1$ should take for the translation followed by a rotation to be an identical transformation to the rotation around (x_r, y_r) can be found as follows.

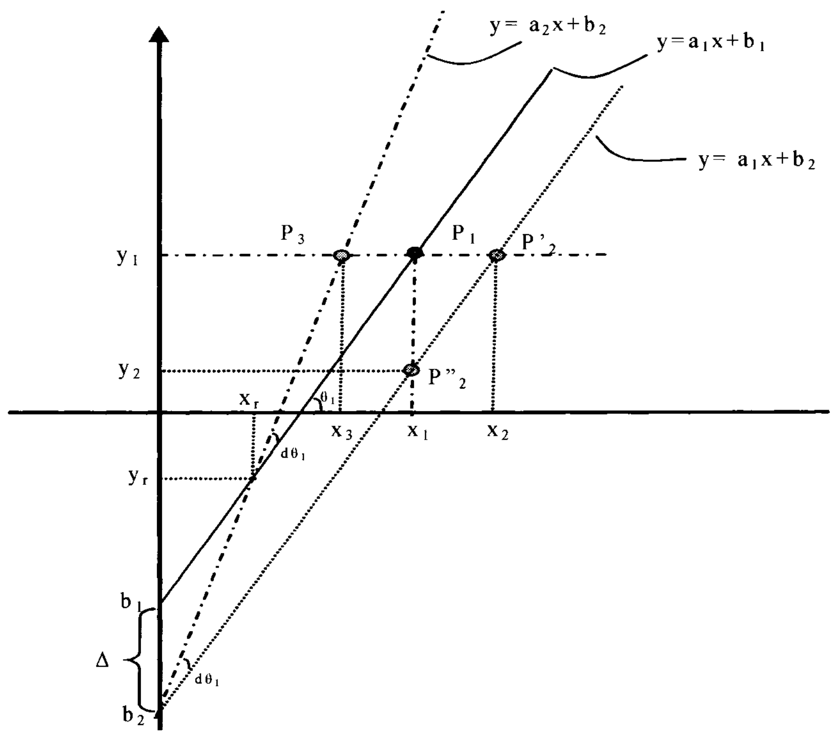

Figure 9.30: Translating and rotating line($y = a_1x + b_1$) by changing the intercept and slope.

$$
\begin{aligned}
a_1 &= \frac{y_r - b_1}{x_r} \\
a_2 &= \frac{y_r - b_2}{x_r} = \frac{y_r - (\Delta + b_1)}{x_r} \\
a_1 - a_2 &= \frac{\Delta}{x_r}
\end{aligned}
$$

When we express $da_1 = \lim_{a_2 \to a_1} (a_2 - a_1)$, then the necessary translation amount Δ would be as below.

$$\Delta = -(da_1)\, x_r$$

From the figure above, it follows that a unit increase in the slope (when rotated around reference point) would be equivalent to decreasing the intercept by x_r and performing pure rotation. ■

Proposition 4.6. *The derivative of objective function with respect to the slope when it is rotated around a reference point (x_r, y_r) can be found through either (117) or (118). (117) and (118) corresponds to the horizontal and vertical representations of TC in (56) and (57), respectively.*

$$\frac{dTC}{da_r} = \int_{y \in Y_{BR}} \left(\frac{\partial TC}{\partial A^x(y)} - \frac{\partial TC}{\partial A^y(x)} \right) \frac{\partial A^x(y)}{\partial a_r} dy \tag{117}$$

$$\frac{dTC}{da_r} = \int_{x \in X_{BR}} \left(\frac{\partial TC}{\partial A^y(x)} - \frac{\partial TC}{\partial A^x(y)} \right) \frac{\partial A^y(x)}{\partial a_r} dx \tag{118}$$

Proof.

We could differentiate total cost (TC) with respect to the slope(a) of BR. Since we are rotating BR around a reference point (x_r, y_r), we will use a_r (instead of a) to distinguish between pure rotation and rotation around a reference point.

$$\frac{dTC}{da_r} = \int_{y\in Y_{BR}} \frac{\partial TC}{\partial A^x(y)} \frac{\partial A^x(y)}{\partial a_r} dy + \int_{x\in X_{BR}} \frac{\partial TC}{\partial A^y(x)} \frac{\partial A^y(x)}{\partial a_r} dx$$

We know from the previous proposition that:

$$\frac{\partial A^y(x)}{\partial a_r} = (-a)\, \frac{\partial A^x(y)}{\partial a_r}$$

Furthermore, we could change the integration variable using the following relation:

$$\begin{aligned} y &= ax + b \\ &\quad thus \\ dy &= a\,(dx) \end{aligned}$$

$$\begin{aligned} \frac{dTC}{da_r} &= \int_{y\in Y_{BR}} \frac{\partial TC}{\partial A^x(y)} \frac{\partial A^x(y)}{\partial a_r} dy + \int_{x\in X_{BR}} \frac{\partial TC}{\partial A^y(x)} \frac{\partial A^y(x)}{\partial a_r} dx \\ &= \int_{y\in Y_{BR}} \left(\frac{\partial TC}{\partial A^x(y)} \frac{\partial A^x(y)}{\partial a_r} + \frac{1}{a} \frac{\partial TC}{\partial A^y(x)} \frac{\partial A^y(x)}{\partial a_r} \right) dy \end{aligned}$$

After substituting $\frac{\partial A^y(x)}{\partial a_r} = (-a)\, \frac{\partial A^x(y)}{\partial a_r}$:

$$\begin{aligned} \frac{dTC}{da_r} &= \int_{y\in Y_{BR}} \left(\frac{\partial TC}{\partial A^x(y)} \frac{\partial A^x(y)}{\partial a_r} + \frac{1}{a} \frac{\partial TC}{\partial A^y(x)} (-a)\, \frac{\partial A^x(y)}{\partial a_r} \right) dy \\ &= \int_{y\in Y_{BR}} \left(\frac{\partial TC}{\partial A^x(y)} - \frac{\partial TC}{\partial A^y(x)} \right) \frac{\partial A^x(y)}{\partial a_r} dy \end{aligned}$$

When we multiply each side with da_r:

$$\begin{aligned} dTC &= \int_{y\in Y_{BR}} \left(\frac{\partial TC}{\partial A^x(y)} - \frac{\partial TC}{\partial A^y(x)}\right) dA^x(y)dy \\ \frac{dTC}{dA^x(y)} &= \frac{\partial TC}{\partial A^x(y)} - \frac{\partial TC}{\partial A^y(x)} \end{aligned} \tag{119}$$

Similarly, we could substitute $\frac{\partial A^x(y)}{\partial a_r} = \left(\frac{-1}{a}\right)\frac{\partial A^y(x)}{\partial a_r}$ and obtain the following.

$$\begin{aligned} \frac{dTC}{da_r} &= \int_{y\in Y_{BR}} \frac{\partial TC}{\partial A^x(y)}\frac{\partial A^x(y)}{\partial a_r}dy + \int_{x\in X_{BR}} \frac{\partial TC}{\partial A^y(x)}\frac{\partial A^y(x)}{\partial a_r}dx \\ &= \int_{x\in X_{BR}} \frac{\partial TC}{\partial A^x(y)}\frac{\partial A^x(y)}{\partial a_r}(a)dx + \int_{x\in X_{BR}} \frac{\partial TC}{\partial A^y(x)}\frac{\partial A^y(x)}{\partial a_r}dx \\ &= \int_{x\in X_{BR}} \left(\frac{\partial TC}{\partial A^x(y)}\frac{\partial A^x(y)}{\partial a_r}(a) + \frac{\partial TC}{\partial A^y(x)}\frac{\partial A^y(x)}{\partial a_r}\right) dx \\ &= \int_{x\in X_{BR}} \left(\frac{\partial TC}{\partial A^x(y)}\left(\frac{-1}{a}\right)\frac{\partial A^y(x)}{\partial a_r}(a) + \frac{\partial TC}{\partial A^y(x)}\frac{\partial A^y(x)}{\partial a_r}\right) dx \\ &= \int_{x\in X_{BR}} \left(\frac{\partial TC}{\partial A^y(x)} - \frac{\partial TC}{\partial A^x(y)}\right)\frac{\partial A^y(x)}{\partial a_r}dx \end{aligned}$$

After multiplying each side with da_r:

$$\begin{aligned} dTC &= \int_{x\in X_{BR}} \left(\frac{\partial TC}{\partial A^y(x)} - \frac{\partial TC}{\partial A^x(y)}\right) dA^y(x)dx \\ \frac{dTC}{dA^y(x)} &= \frac{\partial TC}{\partial A^y(x)} - \frac{\partial TC}{\partial A^x(y)} \end{aligned} \tag{120}$$

■

Proposition 4.7.

The partial derivatives of the objective function with respect to single-dimensional allocation decisions satisfy the following relationship.

$$\left(\frac{\partial TC}{\partial A^x(y)} - \frac{\partial TC}{\partial A^y(x)}\right) = [d_p(\mathbf{x}_1^*, \mathbf{x}) - d_p(\mathbf{x}_2^*, \mathbf{x})]\, D(\mathbf{x}) \tag{121}$$

where $x = (A^x(y), A^y(x)) \in BR$ and for $p = L_2$ and $p = L_2^2$.

Proof.

Note that in Section 4.4.2, we have illustrated the TC could be expressed in terms of either horizontal ($A^x(y)$) or vertical ($A^y(x)$) single dimensional allocation decisions. In either case, partial derivative with respect to other single-dimensional allocation decision would be zero. Recall the expression of TC in terms of horizontal ($A^x(y)$) single dimensional allocation decisions:

$$TC = \int_{y \in Y_{BR}} \int_0^{A^x(y)} d_p(\mathbf{x}_1^*, \mathbf{x}) D(\mathbf{x}) d\mathbf{x} + \int_{y \in Y_{BR}} \int_{A^x(y)}^{M} d_p(\mathbf{x}_2^*, \mathbf{x}) D(\mathbf{x}) d\mathbf{x}$$

When we take the partial derivative of TC with respect to $A^x(y)$, i.e. $\frac{\partial TC}{\partial A^x(y)}$, using the Leibniz's rule.

$$\frac{\partial TC}{\partial A^x(y)} = [d_p(\mathbf{x}_1^*, (A^x(y), y)) - d_p(\mathbf{x}_2^*, (A^x(y), y))]\, D(A^x(y), y)$$

Similarly, when we follow the same steps for the expression of TC in terms of $A^y(x)$, we obtain the other form.

$$\frac{\partial TC}{\partial A^y(x)} = [d_p(\mathbf{x}_2^*, (x, A^y(x))) - d_p(\mathbf{x}_1^*, (x, A^y(x)))]\, D(x, A^y(x))$$

Since either $\frac{\partial TC}{\partial A^y(x)}$ or $\frac{\partial TC}{\partial A^x(y)}$ is non-zero, (121) holds true.

Proposition 4.8. *The partial derivative of the objective function with re-*

spect to single-dimensional allocation decisions satisfy the following relationship when the distance measure is separable, i.e. $d_p(\mathbf{x}_i^*, \mathbf{x}) = d_{p_x}(\mathbf{x}_i^*, \mathbf{x}) + d_{p_y}(\mathbf{x}_i^*, \mathbf{x})$

$$\frac{\partial TC}{\partial A^x(y)} = \left[d_{p_x}(\mathbf{x}_1^*, \mathbf{x}) - d_{p_x}(\mathbf{x}_2^*, \mathbf{x})\right] D(\mathbf{x}) \tag{122}$$

$$\frac{\partial TC}{\partial A^y(x)} = \left[d_{p_y}(\mathbf{x}_1^*, \mathbf{x}) - d_{p_y}(\mathbf{x}_2^*, \mathbf{x})\right] D(\mathbf{x}) \tag{123}$$

where $x = (A^x(y), A^y(x)) \in BR$

Proof.

The proof is along the lines of Proposition 4.7. When we express TC in terms of both the horizontal and vertical single-dimensional allocation decisions and take partial derivative as in the previous proof, we would obtain (122) and (122)

$$\begin{aligned} TC = & \int_{y \in Y_{BR}} \int_0^{A^x(y)} d_{p_x}(\mathbf{x}_1^*, \mathbf{x}) D(\mathbf{x}) d\mathbf{x} + \int_{y \in Y_{BR}} \int_{A^x(y)}^{M} d_{p_x}(\mathbf{x}_2^*, \mathbf{x}) D(\mathbf{x}) d\mathbf{x} \\ & + \int_{x \in X_{A1}} \int_0^{M} d_{p_y}(\mathbf{x}_1^*, \mathbf{x}) D(\mathbf{x}) d\mathbf{x} + \int_{x \in X_{BR}} \int_{A^y(x)}^{M} d_{p_y}(\mathbf{x}_1^*, \mathbf{x}) D(\mathbf{x}) d\mathbf{x} \\ & + \int_{x \in X_{BR}} \int_0^{A^y(x)} d_{p_y}(\mathbf{x}_2^*, \mathbf{x}) D(\mathbf{x}) d\mathbf{x} + \int_{x \in X_{A2}} \int_0^{M} d_{p_y}(\mathbf{x}_2^*, \mathbf{x}) D(\mathbf{x}) d\mathbf{x} \end{aligned} \tag{124}$$

Note that, since $d_p(\mathbf{x}_i^*, \mathbf{x}) = d_{p_x}(\mathbf{x}_i^*, \mathbf{x}) + d_{p_y}(\mathbf{x}_i^*, \mathbf{x})$, the following result still holds.

$$
\begin{aligned}
\left(\frac{\partial TC}{\partial A^x(y)} - \frac{\partial TC}{\partial A^y(x)}\right) &= (122) - (123) \\
&= \left[d_{p_x}(\mathbf{x}_1^*, \mathbf{x}) - d_{p_x}(\mathbf{x}_2^*, \mathbf{x})\right] D(\mathbf{x}) \\
&\quad - \left[d_{p_y}(\mathbf{x}_1^*, \mathbf{x}) - d_{p_y}(\mathbf{x}_2^*, \mathbf{x})\right] D(\mathbf{x}) \\
&= \left[d_p(\mathbf{x}_1^*, \mathbf{x}) - d_p(\mathbf{x}_2^*, \mathbf{x})\right] D(\mathbf{x})
\end{aligned}
$$

■

Proposition 4.9. *Hessian of the TC with respect to the allocation line BR parametrized over its slope (a) and intercept (b), for the cases L_2 and L_2^2, can be found as follows:*

$$
\nabla^2 TC = \begin{bmatrix} \frac{\partial^2 TC}{\partial a^2} & \frac{\partial^2 TC}{\partial a \partial b} \\ \frac{\partial^2 TC}{\partial a \partial b} & \frac{\partial^2 TC}{\partial b^2} \end{bmatrix}
$$

where

$$
\frac{d^2 TC}{da^2} = \int_{y \in Y_{BR}} \begin{bmatrix} \left(\frac{\partial^2 TC}{\partial A^x(y)^2} + (a)\,\frac{\partial^2 TC}{\partial A^y(x)^2}\right)\left(\frac{\partial A^x(y)}{\partial a}\right)^2 + \\ \left(\frac{\partial TC}{\partial A^x(y)} - \frac{\partial TC}{\partial A^y(x)}\right)\frac{\partial^2 A^x(y)}{\partial a^2} \end{bmatrix} dy
$$

$$
\frac{d^2 TC}{db^2} = \int_{y \in Y_{BR}} \left(\frac{\partial^2 TC}{\partial A^x(y)^2} + (a)\,\frac{\partial^2 TC}{\partial A^y(x)^2}\right)\left(\frac{\partial A^x(y)}{\partial b}\right)^2 dy
$$

$$
\frac{\partial^2 TC}{\partial a \partial b} = \int_{y \in Y_{BR}} \begin{bmatrix} \left(\frac{\partial^2 TC}{\partial A^x(y)^2} + (a)\frac{\partial^2 TC}{\partial A^y(x)^2}\right)\frac{\partial A^x(y)}{\partial a}\frac{dA^x(y)}{db} + \\ \left(\frac{\partial TC}{\partial A^x(y)} - \frac{\partial TC}{\partial A^y(x)}\right)\frac{\partial^2 A^x(y)}{\partial a \partial b} \end{bmatrix} dy
$$

where $x = (A^x(y), A^y(x)) \in BR$.
$\frac{\partial TC}{\partial A^x(y)}$, $\frac{\partial TC}{\partial A^y(x)}$, $\frac{\partial^2 TC}{\partial A^x(y)^2}$ and $\frac{\partial^2 TC}{\partial A^y(x)^2}$ can be obtained from (56) and (57). Moreover

derivatives of $A^x(y)$ with respect to slope and intercept are as follows.

$$\begin{aligned}
\frac{\partial A^x(y)}{\partial a} &= -\frac{A^x(y)}{a} \text{ and } \frac{\partial^2 A^x(y)}{\partial a^2} = 2\frac{A^x(y)}{a^2} \\
\frac{\partial A^x(y)}{\partial b} &= -\frac{1}{a} \text{ and } \frac{\partial^2 A^x(y)}{\partial b^2} = 0 \\
\frac{\partial^2 A^x(y)}{\partial a \partial b} &= \frac{1}{a^2}
\end{aligned}$$

Proof.

For both distance measures, $\frac{\partial^2 TC}{\partial A^x(y)^2}$ *and* $\frac{\partial^2 TC}{\partial A^y(x)^2}$ can be obtained as in the proofs of Propositions 4.7. and 4.8. We now prove the terms in the Hessian. For the slope, the second order derivative with respect to the slope $\frac{d^2TC}{da^2}$ can be derived as follows:

$$\frac{dTC}{da} = \int_{y \in Y_{BR}} \left(\frac{\partial TC}{\partial A^x(y)} - \frac{\partial TC}{\partial A^y(x)} \right) \frac{\partial A^x(y)}{\partial a} dy$$

$$\frac{d^2TC}{da^2} = \int_{y \in Y_{BR}} \left[\begin{array}{c} \left(\frac{\partial^2 TC}{\partial A^x(y)^2} \frac{\partial A^x(y)}{\partial a} - \frac{\partial^2 TC}{\partial A^y(x)^2} \frac{\partial A^y(x)}{\partial a} \right) \frac{\partial A^x(y)}{\partial a} + \\ \left(\frac{\partial TC}{\partial A^x(y)} - \frac{\partial TC}{\partial A^y(x)} \right) \frac{\partial^2 A^x(y)}{\partial a^2} \end{array} \right] dy$$

since

$$\frac{dA^y(x)}{da} = (-a)\frac{dA^x(y)}{da}$$

$$\frac{d^2TC}{da^2} = \int_{y \in Y_{BR}} \left[\begin{array}{c} \left(\frac{\partial^2 TC}{\partial A^x(y)^2} + (a) \frac{\partial^2 TC}{\partial A^y(x)^2} \right) \left(\frac{\partial A^x(y)}{\partial a} \right)^2 + \\ \left(\frac{\partial TC}{\partial A^x(y)} - \frac{\partial TC}{\partial A^y(x)} \right) \frac{\partial^2 A^x(y)}{\partial a^2} \end{array} \right] dy$$

where

$$\frac{d^2 A^x(y)}{da^2} = 2\frac{A^x(y)}{a^2}$$

Above result follows from the below:

$$\begin{aligned}
\frac{d^2A^x(y)}{da^2} &= \frac{d\left(\frac{dA^x(y)}{da}\right)}{d} \\
\text{since } \frac{dA^x(y)}{da} &= -\frac{A^x(y)}{a} \\
\frac{d^2A^x(y)}{da^2} &= -\left[\frac{dA^x(y)}{da}a^{-1} + A^x(y)(-1)(a_1^{-2})\right] \\
&= 2\frac{A^x(y)}{a^2}
\end{aligned}$$

Now we will first derive the second order derivative with respect to the intercept $\frac{d^2TC}{db^2}$.

$$\frac{dTC}{db} = \int_{y\in Y_{BR}} \left(\frac{\partial TC}{\partial A^x(y)} - \frac{\partial TC}{\partial A^y(x)}\right)\frac{\partial A^x(y)}{\partial b}dy$$

$$\frac{d^2TC}{db^2} = \int_{y\in Y_{BR}} \left[\begin{array}{c} \left(\frac{\partial^2TC}{\partial A^x(y)^2}\frac{\partial A^x(y)}{\partial b} - \frac{\partial^2TC}{\partial A^y(x)^2}\frac{\partial A^y(x)}{\partial b}\right)\frac{\partial A^x(y)}{\partial b} + \\ \left(\frac{\partial TC}{\partial A^x(y)} - \frac{\partial TC}{\partial A^y(x)}\right)\frac{\partial^2 A^x(y)}{\partial b^2} \end{array}\right] dy$$

Since

$$\frac{dA^y(x)}{db} = (-a)\frac{dA^x(y)}{db}$$

$$\frac{d^2TC}{db^2} = \int_{y\in Y_{BR}} \left[\begin{array}{c} \left(\frac{\partial^2TC}{\partial A^x(y)^2} + (a)\frac{\partial^2TC}{\partial A^y(x)^2}\right)\left(\frac{\partial A^x(y)}{\partial b}\right)^2 + \\ \left(\frac{\partial TC}{\partial A^x(y)} - \frac{\partial TC}{\partial A^y(x)}\right)\frac{\partial^2 A^x(y)}{\partial b^2} \end{array}\right] dy$$

Furthermore second order derivative of $A^x(y)$ with respect to the intercept (b).

$$\frac{d^2A^x(y)}{db^2} = 0$$

Above result follows from

$$\frac{d^2A^x(y)}{db^2} = \frac{d\left(\frac{dA^x(y)}{db}\right)}{db} = \frac{d\left(\frac{-1}{a}\right)}{db} = 0$$

As a result:

$$\frac{\partial^2 TC}{\partial b^2} = \int_{y \in Y_{BR}} \left[\left(\frac{\partial^2 TC}{\partial A^x(y)^2} + (a) \frac{\partial^2 TC}{\partial A^y(x)^2} \right) \left(\frac{\partial A^x(y)}{\partial b} \right)^2 \right] dy$$

Now we will derive the second order partial derivatives with respect to the slope and intercept $\frac{d^2TC}{dadb}$. We use the following first.

$$\frac{dTC}{db} = \int_{y \in Y_{BR}} \left(\frac{\partial TC}{\partial A^x(y)} - \frac{\partial TC}{\partial A^y(x)} \right) \frac{\partial A^x(y)}{\partial b} dy$$

$$\begin{aligned} \frac{d^2TC}{dadb} &= \frac{d\left(\frac{dTC}{da}\right)}{db} \\ &= \int_{y \in Y_{BR}} \left[\begin{array}{c} \left(\frac{\partial^2 TC}{\partial A^x(y)^2} \frac{\partial A^x(y)}{\partial b} - \frac{\partial^2 TC}{\partial A^y(x)^2} \frac{\partial A^y(x)}{\partial b} \right) \frac{\partial A^x(y)}{\partial a} + \\ \left(\frac{\partial TC}{\partial A^x(y)} - \frac{\partial TC}{\partial A^y(x)} \right) \frac{\partial^2 A^x(y)}{\partial a \partial b} \end{array} \right] dy \end{aligned}$$

Since

$$\frac{dA^y(x)}{db} = (-a) \frac{dA^x(y)}{db}$$

$$\frac{\partial^2 TC}{\partial a \partial b} = \int_{y \in Y_{BR}} \left[\begin{array}{c} \left(\frac{\partial^2 TC}{\partial A^x(y)^2} + (a) \frac{\partial^2 TC}{\partial A^y(x)^2} \right) \frac{\partial A^x(y)}{\partial a} \frac{dA^x(y)}{db} + \\ \left(\frac{\partial TC}{\partial A^x(y)} - \frac{\partial TC}{\partial A^y(x)} \right) \frac{\partial^2 A^x(y)}{\partial a \partial b} \end{array} \right] dy$$

$$\frac{d^2 A^x(y)}{dadb} = \frac{1}{a^2}$$

Above result follows from

$$\frac{d^2 A^x(y)}{dadb} = \frac{d\left(\frac{dA^x(y)}{db}\right)}{da} = \frac{d\left(\frac{-1}{a}\right)}{da} = \frac{1}{a^2}$$

Hence the Hessian follows as in proposition. ■

9.3 Appendix 5

Proposition 5.3.

Suppose an edge $\mathbf{e}_{ij}$ *passes through two vertices* $\mathbf{v}_k = (v_k^x, v_k^y)$ *and* $\mathbf{v}_t = (v_t^x, v_t^y)$*. Moving the vertex* $\mathbf{v}_k$ *by increasing* v_k^x *and* v_k^y*, would change the slope* (a_{ij}) *and intercept* (b_{ij}) *of the edge* $\mathbf{e}_{ij}$ *according to the following relations:*

$$\frac{da_{ij}}{dv_k^x} = \frac{a_{ij}^2}{(v_t^y - v_k^y)} \quad \text{and} \quad \frac{da_{ij}}{dv_k^y} = \frac{1}{(v_k^x - v_t^x)}$$

$$\frac{db_{ij}}{dv_k^x} = \frac{a_{ij}^2 v_t^x}{(v_k^y - v_t^y)} \quad \text{and} \quad \frac{db_{ij}}{dv_k^y} = \frac{v_t^x}{(v_t^x - v_k^x)}$$

Proof.

Recall from Proposition 4.3. the following relation for the allocation line, i.e. edge, rotating around a reference point (x_r, y_r).

$$\frac{dA^x(y)}{da_r} = \frac{(y_r - y)}{a^2}$$

$$\frac{dA^y(x)}{da_r} = (x - x_r)$$

Hence, with respect to the notation in the Proposition 5.3, we have the following relations.

$$
\begin{aligned}
(v_k^x, v_k^y) &\equiv (A^x(y), A^y(x)) \\
(v_t^x, v_t^y) &\equiv (x_r, y_r) \\
a_{ij} &\equiv a_r\,(= a)
\end{aligned}
$$

Hence $\frac{da_{ij}}{dv_k^x}$ and $\frac{da_{ij}}{dv_k^y}$ follows directly from the inverted result of Proposition 4.3.

$$\frac{da_{ij}}{dv_k^x} = \frac{a_{ij}^2}{(v_t^y - v_k^y)} \qquad \textit{and} \qquad \frac{da_{ij}}{dv_k^y} = \frac{1}{(v_k^x - v_t^x)}$$

For $\frac{db_{ij}}{dv_k^x}$ and $\frac{db_{ij}}{dv_k^y}$ consider the following equation of the edge e_{ij}.

$$y = a_{ij}x + b_{ij}$$

From this equation, $(da_{ij})\,x = -(db_{ij})$ follows, as in Proposition 4.5. Hence we could substitute $\frac{-(db_{ij})}{x}$ in place of da_{ij} and obtain the following.

$$\frac{db_{ij}}{dv_k^x} = \frac{a_{ij}^2 v_t^x}{(v_k^y - v_t^y)} \qquad \textit{and} \qquad \frac{db_{ij}}{dv_k^y} = \frac{v_t^x}{(v_t^x - v_k^x)}$$

. ■

9.4 Appendix 7

Proposition 7.1

The optimal locations of the two facilities ($\mathbf{x}_1^*$ and $\mathbf{x}_2^*$), *given the allocation decisions, satisfy the following conditions when the distance measure is based on the* **Manhattan − Metric** $(\mathbf{L}_1)$:

$$\int_y \int_{x<x_i^*,\ x\in\mathcal{A}_i} D(\mathbf{x})d\mathbf{x} = \int_y \int_{x\geq x_i^*,\ x\in\mathcal{A}_i} D(\mathbf{x})d\mathbf{x} \qquad \text{for } i=1,2$$

$$\int_x \int_{y<y_i^*,\ y\in\mathcal{A}_i} D(\mathbf{x})d\mathbf{x} = \int_x \int_{y\geq y_i^*,\ y\in\mathcal{A}_i} D(\mathbf{x})d\mathbf{x} \qquad \text{for } i=1,2$$

Proof.

Since the proof is identical for x and $y-$dimensions, we will only prove the optmality condition for x_i. For a given $\mathcal{A}_i$, the total travel in Manhattan-metric can be written as:

$$\int_{\mathcal{A}_i} |x_i - x|\, D(\mathbf{x})dxdy = \int_y \int_{\mathcal{A}_i^x(y)} |x_i - x|\, D(\mathbf{x})dxdy$$

Let's define

$$\text{Define } TC_{\mathcal{A}_i}^x = \text{total travel in x-dimension in } \mathcal{A}_i$$

If we discretize $\left(\int_y\right)$ with $\left(\sum_{k=1}^{m}\right)$, we obtain the following equivalent.

$$\begin{aligned} TC_{\mathcal{A}_i}^x &= \int_y \int_{\mathcal{A}_i^x(y)} |x_i - x|\, D(\mathbf{x})dxdy = \sum_{k=1}^{m} \int_{\mathcal{A}_i^x(y_k)} |x_i - x|\, D(x)dx \\ &= \int_{\mathcal{A}_i^x(y_j)} |x_i - x|\, D(x)dx + \sum_{\substack{k=1 \\ k\neq j}}^{m} \int_{\mathcal{A}_i^x(y_k)} |x_i - x|\, D(x)dx \end{aligned}$$

If we express $\mathcal{A}_i^x(y_t) = B_i^x(y_t) + A_i^x(y_t)$:

$$
\begin{aligned}
&= \int\limits_{B_i^x(y_j)}^{B_i^x(y_j)+A_i^x(y_j)} |x_i - x|\, D(x)dx + \sum_{\substack{k=1\\k\neq j}}^{m} \int\limits_{B_i^x(y_k)}^{B_i^x(y_k)+A_i^x(y_k)} |x_i - x|\, D(x)dx \\
&= \int\limits_{B_i^x(y_j)}^{x_i} (x_i - x)D(x)dx - \int\limits_{x_i}^{B_i^x(y_j)+A_i^x(y_j)} (x_i - x)D(x)dx \\
&\quad + \sum_{\substack{k=1\\k\neq j}}^{m} \left[\int\limits_{B_i^x(y_k)}^{x_i} (x_i - x)\, D(x)dx - \int\limits_{x_i}^{B_i^x(y_k)+A_i^x(y_k)} (x_i - x)\, D(x)dx \right]
\end{aligned}
$$

The optimal location (median in this case) could be found from the first order condition. $\frac{dTC_i}{dx_i} = 0$. Using the Leibniz rule, we obtain the following.

$$
\begin{aligned}
\frac{dTC_{\mathcal{A}_i}^x}{dx_i} &= \int\limits_{B_i^x(y_j)}^{x_i} D(x)dx - \int\limits_{x_i}^{B_i^x(y_j)+A_i^x(y_j)} D(x)dx \\
&\quad + \sum_{\substack{k=1\\k\neq j}}^{m} \left[\int\limits_{B_i^x(y_k)}^{x_i} D(x)dx - \int\limits_{x_i}^{B_i^x(y_k)+A_i^x(y_k)} D(x)dx \right] = 0
\end{aligned}
$$

After combining positive and negative signed terms and taking $m \to \infty$,

$$
\begin{aligned}
\frac{dTC_{\mathcal{A}_i}^x}{dx_i} &= \int_y \int\limits_{B_i^x(y_j)}^{x_i} D(x)dx - \int_y \int\limits_{x_i}^{B_i^x(y_j)+A_i^x(y_j)} D(x)dx \\
&= \int_y \left[\int\limits_{x<x_i, x\in \mathcal{A}_i^x(y)} D(\mathbf{x})dx - \int\limits_{x\geq x_i, x\in \mathcal{A}_i^x(y)} D(\mathbf{x})dx \right] dy \underset{x_i = x_i^*}{=} 0
\end{aligned}
$$

Hence the proof is complete. ■

Proposition 7.2

The partial derivative of the objective function TC with respect to p1, p2 and p3 is as follows:

$$
\begin{aligned}
\frac{dTC}{dp_1} &= \int_{y \in Y_{BR1}} |x_1 - p_1| \, D(p_1, y)dy - \int_{y \in Y_{BR1}} |x_2 - p_1| \, D(p_1, y)dy \\
&+ \int_0^M |y_1 - y| \, D(p_1, y)dy - \int_{p_1+p_3}^M |y_1 - y| \, D(p_1, y)dy \\
&- \int_0^{p_1+p_3} |y_2 - y| \, D(p_1, y)dy
\end{aligned}
$$

$$
\begin{aligned}
\frac{dTC}{dp_2} &= \int_{y \in Y_{BR3}} |x_1 - p_2| \, D(p_2, y)dy - \int_{y \in Y_{BR3}} |x_2 - p_2| \, D(p_2, y)dy \\
&- \int_0^M |y_2 - y| \, D(p_2, y)dy + \int_{p_2+p_3}^M |y_1 - y| \, D(p_2, y)dy \\
&+ \int_0^{p_2+p_3} |y_2 - y| \, D(p_2, y)dy
\end{aligned}
$$

$$
\begin{aligned}
\frac{dTC}{dp_3} &= -\int_{y \in Y_{BR2}} |x_1 - (y - p_3)| \, D(y - p_3, y)dy \\
&+ \int_{y \in Y_{BR2}} |x_2 - (y - p_3)| \, D(y - p_3, y)dy \\
&- \int_{p_1}^{p_2} |y_1 - (x + p_3)| \, D(x, (x + p_3))dx \\
&+ \int_{p_1}^{p_2} |y_2 - (x + p_3)| \, D(x, (x + p_3))dx
\end{aligned}
$$

Proof.

Consider Figure 7.11. Here we will use the triple-tuplet (p_1, p_2, p_3) to characterize this special allocation line.

$$br(y)^{-1} = \left\{ \begin{array}{cc} p_1 & y \leq p_1 + p_3 \\ y - p_3 & p_1 + p_3 < y \leq p_2 + p_3 \\ p_2 & y > p_2 + p_3 \end{array} \right\}$$

$$br(x) = \left\{ \begin{array}{cc} 0 & x \leq p_1 \\ x + p_3 & p_1 < x \leq p_2 \\ M & x > p_2 \end{array} \right\}$$

Recall from Chapter 7, that $A^y(x) = br(x)$ and $A^x(y) = br(y)^{-1}$ and when we substitute the expression for these single dimensional allocation decisions.

$$\begin{aligned} TC &= \int_{y \in Y_{BR1}} \int_0^{p_1} |x_1 - x| \, D(x,y) dx dy + \int_{y \in Y_{BR2}} \int_0^{y-p_3} |x_1 - x| \, D(x,y) dx dy \\ &+ \int_{y \in Y_{BR3}} \int_0^{p_2} |x_1 - x| \, D(x,y) dx dy + \int_{y \in Y_{BR1}} \int_{p_1}^{M} |x_2 - x| \, D(x,y) dx dy \\ &+ \int_{y \in Y_{BR2}} \int_{y-p_3}^{M} |x_2 - x| \, D(x,y) dx dy + \int_{y \in Y_{BR3}} \int_{p_2}^{M} |x_2 - x| \, D(x,y) dx dy \\ &+ \int_0^{p_1} \int_0^{M} |y_1 - y| \, D(x,y) dy dx + \int_{p_1}^{p_2} \int_{x+p_3}^{M} |y_1 - y| \, D(x,y) dy dx \\ &+ \int_{p_1}^{p_2} \int_0^{x+p_3} |y_2 - y| \, D(x,y) dy dx + \int_{p_2}^{M} \int_0^{M} |y_2 - y| \, D(x,y) dy dx \end{aligned}$$

When we apply the Leibniz rule to obtain the derivatives $\frac{dTC}{dp_1}$, $\frac{dTC}{dp_2}$ and $\frac{dTC}{dp_3}$, proposition results follows. ■

Proposition 7.3

The differential change in the optimal locations with respect to single dimensional allocation decisions satisfy the following conditions when the distance *measure is based on the* **Manhattan – Metric** $(\mathbf{L}_1)$:

$$\begin{aligned}
\frac{\partial x_i^*}{\partial A_i^x(y)} &= \frac{D(B_i^x(y)+A_i^x(y),y)}{2D(x_i^*,y)} \qquad \text{for } i=1,2,...,n \\
\frac{\partial y_i^*}{\partial A_i^y(x)} &= \frac{D(x,B_i^y(x)+A_i^y(x))}{2D(x,y_i^*)} \qquad \text{for } i=1,2,...,n
\end{aligned}$$

$$\begin{aligned}
\frac{\partial x_i^*}{\partial A_{i-1}^x(y)} &= \frac{D(B_i^x(y),y)}{2D(x_i^*,y)} \qquad \text{for } i=1,2,...,n \\
\frac{\partial y_i^*}{\partial A_{i-1}^y(x)} &= \frac{D(x,B_i^y(x))}{2D(x,y_i^*)} \qquad \text{for } i=1,2,...,n
\end{aligned}$$

Proof.

For brevity, we will outline the proof which is based on application of the Leibniz's rule. Recall from proposition 7.1 the following first-order condition:

$$\int_y \int_{x<x_i^*,\ x\in\mathcal{A}_i} D(\mathbf{x})d\mathbf{x} - \int_y \int_{x\geq x_i^*,\ x\in\mathcal{A}_i} D(\mathbf{x})d\mathbf{x} = 0$$

When we re-express above as follows and take the partial derivative with respect to $A_i^x(y)$. Note that we will consider x_i^* as a function of $A_i^x(y)$, i.e. $x_i^*\left(A_i^x(y)\right)$.

$$\int_y \int_{x=B_i^x(y)}^{x_i^*} D(\mathbf{x})d\mathbf{x} - \int_y \int_{x=x_i^*}^{x=B_i^x(y)+A_i^x(y)} D(\mathbf{x})d\mathbf{x} = 0$$

Then, below follows.

$$\frac{\partial x_i^*}{\partial A_i^x(y)} = \frac{D(B_i^x(y)+A_i^x(y),y)}{2D(x_i^*,y)}$$

For the $\frac{\partial x_i^*}{\partial A_{i-1}^x(y)}$, we will express $B_i^x(y)$ as $B_i^x(y) = B_{i-1}^x(y) + A_{i-1}^x(y)$ and apply the Leibniz's rule to differentiate the equation below with respect to $A_{i-1}^x(y)$.

$$\int_y \int_{x=B_{i-1}^x(y)+A_{i-1}^x(y)}^{x_i^*} D(\mathbf{x})d\mathbf{x} - \int_y \int_{x=x_i^*}^{x=B_{i-1}^x(y)+A_{i-1}^x(y)+A_i^x(y)} D(\mathbf{x})d\mathbf{x} = 0$$

Hence, the differential relation between the optimal location and its preceding single-dimensional allocation decisions follows as below.

$$\frac{\partial x_i^*}{\partial A_{i-1}^x(y)} = \frac{D(B_i^x(y), y)}{2D(x_i^*, y)}$$

Similar procedure applies for $\frac{\partial y_i^*}{\partial A_i^y(x)}$ and $\frac{\partial y_i^*}{\partial A_{i-1}^y(x)}$.

10 REFERENCES

References

[1] Agarwal, P. K., and Sharir, M. (1998). "Efficient algorithms for geometric optimization." ACM Computing Surveys, 30(4), 412-458.

[2] Avella, P., Benati, S., Martinez, L. C., Dalby, K., Di Girolamo, D., Dimitrijevic, B., Ghiani, G., Giannikos, I., Guttmann, N., Hultberg, T. H., Fliege, J., Marin, A., Marquez, M. M., Ndiaye, M. M., Nickel, S., Peeters, P., Brito, D. P., Policastro, S., de Gama, F. A. S., and Zidda, P. (1998). "Some personal views on the current state and the future of Locational Analysis." European Journal of Operational Research, 104(2), 269-287.

[3] Aykin, T., and Babu, A. J. G. (1987). "Constrained Large-Region Multifacility Location-Problems." Journal of the Operational Research Society, 38(3), 241-252.

[4] Aykin, T., and Brown, G. F. (1992). "Interacting New Facilities and Location-Allocation Problems." Transportation Science, 26(3), 212-222.

[5] Balas, E., and Yu, C. S. (1982). " A note on the Weiszfeld-Kuhn algorithm for the General Fermat Problem, Management Sciences Research Report ", CARNEGIE-MELLON UNIV PITTSBURGH PA, 1-6.

[6] Bazaraa, M. S., Sherali, H. D., and Shetty, C. M. (1993). Nonlinear Programming : Theory and Algorithms, Wiley, New York.

[7] Beckmann, M. J., and Thisse, J. F. (1986). "The location of production activities." Handbook of Regional and Urban Economics, P. Nijkamp, ed., North-Holland, Amsterdam, 21-95.

[8] Bennett, C. D., and Mirakhor, A. (1974). "Optimal facility location with respect to several service regions." Journal of Regional Science, 14, 131-136.

[9] Berman, O. (1997). "Deterministic flow-demand location problems." Journal of the Operational Research Society, 48(1), 75-81.

[10] Berman, O., Drezner, Z., and Wesolowsky, G. O. (2002). "Satisfying partial demand in facilities location." IIE Transactions, 34(11), 971-978.

[11] Berman, O. and Krass, D. (1998). "Flow intercepting spatial interaction model: a new approach to optimal location of competitive facilities." Location Science, 6(1), 41-65.

[12] Berman, O., and Parkan, C. (1984) "Sequential facility location with distance dependent demand." Journal of Operations Management, 3, 261-268

[13] Bongartz, I., Calamai, P. H., and Conn, A. R. (1994). "A Projection Method for L(P) Norm Location-Allocation Problems." Mathematical Programming, 66(3), 283-312.

[14] Brandeau, M. L., and Chiu, S. S. (1989). "An Overview of Representative Problems in Location Research." Management Science, 35(6), 645-674.

[15] Brimberg, J. (1995). "The Fermat-Weber location problem revisited." Mathematical Programming, 71(1), 71-76.

[16] Brimberg, J., Hansen, P., Mlandinovic, N., and Taillard, E. (2000). "Improvement and Comparison of Heuristics for Solving the Uncapacitated Multisource Weber Problem." Operations Research, 48(3), 444-460.

[17] Brimberg, J., Kakhki, H. T., and Wesolowsky, G. O. (2003). "Location among regions with varying norms." Annals of Operations Research, 122(1-4), 87-102.

[18] Brimberg, J., and Love, R. F. (1991). "Estimating Travel Distances by the Weighted Lp Norm." Naval Research Logistics, 38(2), 241-259.

[19] Brimberg, J., and Love, R. F. (1992). "Local convergence in a generalized Fermat-Weber problem." Annals of Operations Research, 40(1), 33-66.

[20] Brimberg, J., and Love, R. F. (1993). "Global Convergence of a Generalized Iterative Procedure for the Minisum Location Problem with L(P) Distances." Operations Research, 41(6), 1153-1163.

[21] Brimberg, J., and Love, R. F. (1995a). "Generalized Hull Properties for Location-Problems." IIE Transactions, 27(2), 226-232.

[22] Brimberg, J., and Love, R. F. (1995b). "Properties of Ordinary and Weighted Sums of Order-P Used for Distance." Rairo-Recherche Operationnelle-Operations Research, 29(1), 59-72.

[23] Brimberg, J., Love, R. F., and Walker, J. H. (1995). "The Effect of Axis Rotation on Distance Estimation." European Journal of Operational Research, 80(2), 357-364.

[24] Brimberg, J., Love, R. F., (1998). "Solving a Class of Two-Dimensional Uncapacitated Location-Allocation Problems by Dynamic Programming." Operations Research, 46(5), 702-709

[25] Brimberg, J., and Mladenovic, N. (1996a). " Solving the continuous location-allocation problem with tabu search." Studies in Locational Analysis, 8, 23-32.

[26] Brimberg, J., and Mladenovic, N. (1996b). "A variable neighbourhood algorithm for solving the continuous location-allocation problem." Studies in Locational Analysis, 10(1-12).

[27] Brimberg, J., Mladenovic, N., and Salhi, S. (2004). "The multi-source Weber problem with constant opening cost." Journal of the Operational Research Society, 55(6), 640-646.

[28] Brimberg, J., and Salhi, S. (2005). "A continuous location-allocation problem with zone-dependent fixed cost." Annals of Operations Research, 136(1), 99-115.

[29] Brimberg, J., and Wesolowsky, G. O. (2000). "Note: Facility location with closest rectangular distances." Naval Research Logistics, 47(1), 77-84.

[30] Brimberg, J., and Wesolowsky, G. O. (2002). "Minisum location with closest Euclidean distances." Annals of Operations Research, 111(1-4), 151-165.

[31] Cabot, A. V., Francis, R. L., and Stary, M. A. (1970). "A network flow solution to a rectilinear distance facility location problem." AIIE Transactions, 2, 132-141.

[32] Calamai, P. H., and Conn, A. R. (1980). "A Stable Algorithm for Solving the Multifacility Location Problem Involving Euclidean Distances." SIAM Journal on Scientific and Statistical Computing, 1(4), 512-526.

[33] Calamai, P. H., and Conn, A. R. (1987). "A Projected Newton Method for Lp Norm Location-Problems." Mathematical Programming, 38(1), 75-109.

[34] Campbell, J. F. (1992). "Location-Allocation for Distribution to a Uniform Demand with Transshipments." Naval Research Logistics, 39(5), 635-649.

[35] Carrizosa, E., Munoz-Marquez, M., and Puerto, J. (1998). "The Weber problem with regional demand." European Journal of Operational Research, 104(2), 358-365.

[36] Cavalier, T. M., and Sherali, H. D. (1986). "Euclidean Distance Location-Allocation Problems with Uniform Demands over Convex Polygons." Transportation Science, 20(2), 107-116.

[37] Chandrasekaran, R., and Tamir, A. (1989). "Open Questions Concerning Weiszfeld Algorithm for the Fermat-Weber Location Problem." Mathematical Programming, 44(3), 293-295.

[38] Charalambous, C. (1985). "Acceleration of the Hap Approach for the Multifacility Location Problem." Naval Research Logistics, 32(3), 373-389.

[39] Chen, P. C., Hansen, P., Jaumard, B., and Tuy, H. (1998). "Solution of the multisource weber and conditional weber problems by d.-c. programming." Operations Research, 46(4), 548-562.

[40] Chen, R. (1983). "Solution of Minisum and Minimax Location-Allocation Problems with Euclidean Distances." Naval Research Logistics, 30(3), 449-459.

[41] Cheung, T. Y. (1980). "Multi-Facility Location Problem with Rectilinear Distance by the Minimum-Cut Approach." Acm Transactions on Mathematical Software, 6(3), 387-390.

[42] Cooper, L. (1963). "Location-Allocation Problems " Operations Research, 11(3), 331-343.

[43] Cooper, L. (1964). "Heuristic models for location-allocation problems." SIAM REVIEW, 6(1), 37-52.

[44] Cooper, L. (1972). "The Transportation-Location Problem " Operations Research, 20(1), 94-108.

[45] Cooper, L., and Katz, I. N. (1981). "The Weber Problem Revisited." Computers & Mathematics with Applications, 7(3), 225-234.

[46] Daganzo, C. F. (1991). Logistics Systems Analysis, Berlin.

[47] Dasci, A. (2001). "Discrete and Continuous models for production-distribution systems," PhD Thesis, Faculty of Management, McGill University, Montreal.

[48] Dasci, A., and Laporte, G. (2004). "Location and pricing decisions of a multistore monopoly in a spatial market." Journal of Regional Science, 44(3), 489-515.

[49] Dasci, A., and Laporte, G. (2005a). "An analytical approach to the facility location and capacity acquisition problem under demand uncertainty." Journal of the Operational Research Society, 56(4), 397-405.

[50] Dasci, A., and Laporte, G. (2005b). "A continuous model for multistore competitive location." Operations Research, 53(2), 263-280.

[51] Dasci, A., and Verter, V. (2001). "A continuous model for production-distribution system design." European Journal of Operational Research, 129(2), 287-298.

[52] Dasci, A., and Verter, V. (2005). "Evaluation of plant focus strategies: A continuous approximation framework." Annals of Operations Research, 136(1), 303-327.

[53] Denardo, E. V., Huberman, G., and Rothblum, U. G. (1982). "Optimal Locations on a Line Are Interleaved " Operations Research, 30(4), 745-759.

[54] Dowling, P. D., and Love, R. F. (1986). "Bounding Methods for Facilities Location Algorithms." Naval Research Logistics, 33(4), 775-787.

[55] Dowling, P. D., and Love, R. F. (1987). "An Evaluation of the Dual as a Lower Bound in Facilities Location-Problems." IIE Transactions, 19(2), 160-166.

[56] Drezner, T., and Drezner, Z. (1997). "Replacing continuous demand with discrete demand in a competitive location model." Naval Research Logistics, 44(1), 81-95.

[57] Drezner, Z. (1982). "Competitive Location Strategies for 2 Facilities." Regional Science and Urban Economics, 12(4), 485-493.

[58] Drezner, Z. (1984). "The Planar 2-Center and 2-Median Problems." Transportation Science, 18(4), 351-361.

[59] Drezner, Z. (1985). "A Solution to the Weber Location Problem on the Sphere " Journal of the Operational Research Society, 36(4), 333-334.

[60] Drezner, Z. (1986). "Location of Regional Facilities." Naval Research Logistics Quarterly, 33, 523-529.

[61] Drezner, Z. (1992). "A note on the Weber location problem." Annals of Operations Research, 40, 153-161.

[62] Drezner, Z. (1995). Facility Location : A Survey of Applications and Methods, Springer.

[63] Drezner, Z., and Guyse, J. (1999). "Application of decision analysis techniques to the Weber facility location problem." European Journal of Operational Research, 116(1), 69-79.

[64] Drezner, Z., and Weslowsky, G. O. (1980). "Optimal Location of a Facility Relative to Area Demands." Naval Research Logistics, 27(2), 199-206.

[65] Drezner, Z., and Wesolowsky, G. O. (1978a). "Facility Location on a Sphere " Journal of the Operational Research Society, 29(10), 997-1004.

[66] Drezner, Z., and Wesolowsky, G. O. (1978b). "A Trajectory Method for the Optimization of the Multi-Facility Location Problem with lp Distances " Management Science, 24(14), 1507-1514.

[67] Drezner, Z., and Wesolowsky, G. O. (1989). "Multi-Buyer Discount Pricing." European Journal of Operational Research, 40(1), 38-42.

[68] Drezner, Z., and Wesolowsky, G. O. (1996). "Location-allocation on a line with demand-dependent costs." European Journal of Operational Research, 90(3), 444-450.

[69] Drezner, Z., and Wesolowsky, G. O. (1999a). "Allocation of demand when cost is demand-dependent." Computers & Operations Research, 26(1), 1-15.

[70] Drezner, Z., and Wesolowsky, G. O. (1999b). "Allocation of discrete demand with changing costs." Computers & Operations Research, 26(14), 1335-1349.

[71] Eaton, B. C., and Lipsey, R. G. (1975). " The Principle of Minimum Differentiation Reconsidered: Some New Developments in the Theory of Spatial Competition." Review of Economic Studies, 42(1), 27-49.

[72] Elzinga, D. J., and Hearn, D. W. (1983). "On Stopping Rules for Facilities Location Algorithms." IIE Transactions, 15(1), 81-83.

[73] Erlebacher, S. J., and Meller, R. D. (2000). "The interaction of location and inventory in designing distribution systems." IIE Transactions, 32(2), 155-166.

[74] Erlenkotter, D. (1989). "The general market area model." Annals of Operations Research, 18, 45-70.

[75] Eyster, J. W., and White, J. A. (1973). "Some Properties of the Squared -Euclidean Distance Location Problem." AIIE Transactions, 5(3), 276-280.

[76] Eyster, J. W., White, J. A., and Wierwille, W. W. (1973). "On Solving Multifacility Location Problems Using a Hyperboloid Approximation Procedure." AIIE Transactions, 5, 1-6.

[77] Fekete, S. P., Mitchell, J. S. B., and Beurer, K. (2005). "On the continuous Fermat-Weber problem." Operations Research, 53(1), 61-76.

[78] Fliege, J., and Nickel, S. (2000). "An Interior Point Method for Multifacility Location Problems with Forbidden Regions." Studies in Locational Analysis, 14, 23-46.

[79] Francis, R. L. (1963). "A Note on the Optimum Location of New Machines in Existing Plant Location." AIIE Transactions, 14(1), 57-59.

[80] Francis, R. L., and Cabot, A. V. (1972). "Properties of a multi-facility location problem involving Euclidean distances" Naval Research Logistics Quarterly, 19, 335-353.

[81] Francis, R. L., Lowe, T. J., and Tamir, A. (2000). "Aggregation error bounds for a class of location models." Operations Research, 48(2), 294-307.

[82] Francis, R. L., Mcginnis, L. F., and White, J. A. (1983). "Locational Analysis." European Journal of Operational Research, 12(3), 220-252.

[83] Francis, R. L., and White, J. A. (1998). Facility Layout and Location: An Analytical Approach, Prentice-Hall Inc.

[84] Gamal, M. D. H., and Salhi, S. (2001). "Constructive heuristics for the uncapacitated continuous location-allocation problem." Journal of the Operational Research Society, 52(7), 821-829.

[85] Geoffrion, A. M. (1979). "Making better use of optimization capability in distribution system planning." AIIE Transactions, 11, 96-108.

[86] Goodchild, M. F. (1979). "The aggregation problem in Location-allocation " Geographical Analysis 11, 240-255.

[87] Hale, T. S., and Moberg, C. R. (2003). "Location science research: A review." Annals of Operations Research, 123(1-4), 21-35.

[88] Hamacher, H. W., and Klamroth, K. (2000). "Planar Weber location problems with barriers and block norms." Annals of Operations Research 96(1-4), 191-208.

[89] Hamacher, H. W., and Nickel, S. (1995). "Restricted Planar Location Problems and Applications." Naval Research Logistics, 42(6), 967-992.

[90] Hamacher, H. W., and Nickel, S. (1998). "Classification of Location Problems." Location Science, 6, 229-242.

[91] Hansen, P., Mladenovic, N., and Taillard, E. (1998). "Heuristic solution of the multisource Weber problem as a p-median problem." Operations Research Letters, 22(2-3), 55-62.

[92] Hansen, P., Peeters, D., Richard, D., and Thisse, J. F. (1985). "The Minisum and Minimax Location-Problems Revisited." Operations Research, 33(6), 1251-1265.

[93] Hansen, P., Perreur, J., and Thisse, J. F. (1980). "Location Theory, Dominance, and Convexity - Some Further Results." Operations Research, 28(5), 1241-1250.

[94] Holzapfel, R. P. (1986). Geometry and Arithmetic around Euler Partial Differential Equations, Reidel, Dordrecht.

[95] Houck, C. R., Joines, J. A., and Kay, M. G. (1996). "Comparison of genetic algorithms, random restart and two-opt switching for solving large location-allocation problems." Computers & Operations Research, 23(6), 587-596.

[96] Houck, C. R., Joines, J. A., and Kay, M. G. (2006). "Characterizing search spaces for Tabu search." Manuscript, Department of Industrial Engineering, North Carolina State University, 1-21.

[97] Huff, D. L. (1964). "Defining and Estimating a Trading Area." Journal of Marketing, 28, 34-38.

[98] Iri, M., Murota, K., and Ohya, T. (1983). "A fast Voronoi-diagram algorithm with applications to geographical optimization." Eleventh IFIP Conference on System Modelling and Optimization, 273-288.

[99] Juel, H. (1984). "On a Rational Stopping Rule for Facilities Location Algorithms." Naval Research Logistics, 31(1), 9-11.

[100] Juel, H., and Love, R. F. (1976). "An Efficient Computational Procedure for Solving the Multi-Facility Rectilinear Facilities Location Problem " Operational Research Quarterly, 27(3), 697-703.

[101] Juel, H., and Love, R. F. (1983). "Hull Properties in Location-Problems." European Journal of Operational Research, 12(3), 262-265.

[102] Kafer, B., and Nickel, S. (2001). "Error bounds for the approximative solution of restricted planar location problems." European Journal of Operational Research, 135(1), 67-85.

[103] Katz, I. N. (1974). "Local convergence in Fermat's problem." Mathematical Programming, 6(1), 89-104.

[104] Katz, I. N., and Cooper, L. (1980). "Optimal Location on a sphere." Computers and Mathematics with Applications, 6, 175-196.

[105] Kimball, W. S. (1952). Calculus of Variations by Parallel Displacement, Butterworth, London.

[106] Kolen, A. (1981). "Equivalence between the Direct Search Approach and the Cut Approach to the Rectilinear Distance Location Problem." Operations Research, 29(3), 616-620.

[107] Koshizuka, T., and Kurita, O. (1991). "Approximate Formulas of Average Distances Associated with Regions and Their Applications to Location-Problems." Mathematical Programming, 52(1), 99-123.

[108] Kuenne, R. E., and Soland, R. M. (1972). "Exact and approximate solutions to the multisource weber problem." Mathematical Programming, 3(1), 193-209.

[109] Kuhn, H. W. (1973). "A note on Fermat's problem." Mathematical Programming, 4(1), 98-107.

[110] Kuhn, H. W., and Kuenne, R. E. (1962). "An efficient algorithm for the numerical solution of the generalized Weber problem in spatial economics." Journal of Regional Science, 4(2), 21-33.

[111] Langevin, A., Mbaraga, P., and Campbell, J. F. (1996). "Continuous approximation models in freight distribution: An overview." Transportation Research Part B-Methodological, 30(3), 163-188.

[112] Leamer, E. E. (1968). "Locational equilibria." Journal of Regional Science, 8, 229-242.

[113] Leonard, D., and Ngo, V. D. (1992). Optimal Control Theory and Static Optimization in Economics, Cambridge University Press, Cambridge.

[114] Levin, Y., and Ben-Israel, A. (2002). "The Newton Bracketing method for convex minimization." Computational Optimization and Applications, 21(2), 213-229.

[115] Levin, Y., and Ben-Israel, A. (2004). "A heuristic method for large-scale multi-facility location problems." Computers & Operations Research, 31(2), 257-272.

[116] Li, Y. Y. (1998). "A Newton acceleration of the Weiszfeld algorithm for minimizing the sum of Euclidean distances." Computational Optimization and Applications, 10(3), 219-242.

[117] Lindelöw, M. and A.Wagstaff, (2003). "Health Facility Surveys: An Introduction". The World Bank, World Bank Policy Research Working Paper 2953, 1-50.

[118] Love, R. F. (1972). "A computational procedure for optimally locating a facility with respect to several rectangular regions." Journal of Regional Science, 12, 233-242.

[119] Love, R. F. (1974). "The Dual of a Hyperbolic Approximation to the Generalized Constrained Multi-Facility Location Problem with Lp Distances " Management Science, 21(1), 22-33.

[120] Love, R. F. (1976). "One-Dimensional Facility Location-Allocation Using Dynamic Programming " Management Science, 22(5), 614-617.

[121] Love, R. F., and Dowling, P. D. (1989a). "A Generalized Bounding Method for Multifacility Location Models." Operations Research, 37(4), 653-657.

[122] Love, R. F., and Dowling, P. D. (1989b). "A new bounding method for single facility location models." Annals of Operations Research, 18(1), 103-112.

[123] Love, R. F., and Juel, H. (1982). "Properties and Solution Methods for Large Location Allocation Problems." Journal of the Operational Research Society, 33(5), 443-452.

[124] Love, R. F., and Morris, J. G. (1975). "Solving Constrained Multi-Facility Location Problems Involving l_{p} Distances Using Convex Programming " Operations Research, 23(3), 581-587.

[125] Love, R. F., and Yeong, W. Y. (1981). "A Stopping Rule for Facilities Location Algorithms." Aiie Transactions, 13(4), 357-362.

[126] Luenberger, D. (1984). Linear and Nonlinear Programming, Addison-Wesley Inc., Reading, Massachusetts.

[127] Maranzana, F. E. (1964). "On the Location of Supply Points to Minimize Transport Costs." Operations Research, 15(3), 261-270.

[128] Maruchek, A. S., and Aly, A. A. (1981). "An Efficient Algorithm for the Location-Allocation Problem with Rectangular Regions." Naval Research Logistics, 28(2), 309-323.

[129] Megiddo, N., and Supowit, K. J. (1984). "On the Complexity of Some Common Geometric Location-Problems." Siam Journal on Computing, 13(1), 182-196.

[130] Miehle, E. (1958). "Link-length minimization in networks." Operations Research, 6, 232-243.

[131] Miller, H. J. (1996). "GIS and geometric representation in facility location problems." International Journal of Geographical Information Systems, 10(7), 791-816.

[132] Morris, J. G. (1981). "Convergence of the Weiszfeld Algorithm for Weber Problems Using a Generalized Distance Function." Operations Research, 29(1), 37-48.

[133] Morris, J. G., and Verdini, W. A. (1979). "Minisum Ip Distance Location-Problems Solved Via a Perturbed Problem and Weiszfelds Algorithm." Operations Research, 27(6), 1180-1188.

[134] Murtagh, B. A., and Niwattisyawong, S. R. (1982). "An Efficient Method for the Multi-Depot Location Allocation Problem." Journal of the Operational Research Society, 33(7), 629-634.

[135] Newell, G. F. (1973). "Scheduling, Location, Transportation, and Continuum Mechanics; Some Simple Approximations to Optimization Problems." SIAM Journal on Applied Mathematics, 25(3), 346-360.

[136] Newling, B. E. (1969). "The spatial variation of urban population densities." Geographical Review, 59, 242-252.

[137] Norman, S. K., Rogers, D. F., and Levy, M. S. (1999). "Error bound comparisons for aggregation disaggregation techniques applied to the transportation problem." Computers & Operations Research, 26(10-11), 1003-1014.

[138] Novaes, A. G. N., de Cursi, J. E. S., and Graciolli, O. D. (2000). "A continuous approach to the design of physical distribution systems." Computers & Operations Research, 27(9), 877-893.

[139] Odoni, A. R., and Sadiq, G. (1982). "2 Planar Facility Location-Problems with High-Speed Corridors and Continuous Demand." Regional Science and Urban Economics, 12(4), 467-484.

[140] Ohya, T., Iri, M., and Murota, K. (1984). "Improvements of the Incremental Method for the Voronoi Diagram with Computational Compari-

son of Various Algorithms." Journal of the Operations Research Society of Japan, 27(4), 306-337.

[141] Okabe, A., Boots, B., and Sugihara, K. (2000). Spatial tessellations, J.Wiley & Sons, Chichester.

[142] Okabe, A., Okunuki, K., and Suzuki, A. (1998). "A computational method for optimizing the hierarchy and spatial configuration of successively inclusive facilities on a continuous plane." Location Science, 5(4), 255-268.

[143] Okabe, A., and Suzuki, A. (1997). "Locational optimization problems solved through Voronoi diagrams." European Journal of Operational Research, 98(3), 445-456.

[144] Okelly, M. E. (1986). "The Location of Interacting Hub Facilities." Transportation Science, 20(2), 92-106.

[145] Ostresh, L. M. (1975). "An efficient algorithm for solving two center location-allocation problem." Journal of Regional Science, 15, 209-216.

[146] Ostresh, L. M. (1977). "The multifacIlity problem: Applications and descent theorems." Journal of Regional Science, 17, 409-419.

[147] Ostresh, L. M. (1978). "On the Convergence of a Class of Iterative Methods for Solving the Weber Location Problem." Operations Research, 26(4), 697-609.

[148] Ouyang, Y., and Daganzo, C. F. (2006). "Discretization and validation of the continuum approximation scheme for terminal system design." Transportation Science, 40(1), 89-98.

[149] Overton, M. L. (1983). "A Quadratically Convergent Method for Minimizing a Sum of Euclidean Norms." Mathematical Programming, 27(1), 34-63.

[150] Picard, H., and Ratliff, H. D. (1978). "A Cut Approach to the Rectilinear Distance Facility Location Problem." Operations Research, 26(3), 422-433.

[151] Plastria, F. (1987). "Solving General Continuous Single Facility Location-Problems by Cutting Planes." European Journal of Operational Research, 29(1), 98-110.

[152] Plastria, F. (1992). "When Facilities Coincide - Exact Optimality Conditions in Multifacility Location." Journal of Mathematical Analysis and Applications, 169(2), 476-498.

[153] Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. (1988). Numerical Recipes in C, Cambridge University Press, Cambridge, UK.

[154] Pritsker, A. A. B. (1973). "A Note to Correct the Procedure of Pritsker and Ghare for Locating New Facilities with Respect to Existing Facilities." AIIE Transactions, 5, 84-86.

[155] Pritsker, A. A. B., and Ghare, P. M. (1970). "Locating New Facilities with Respect to Existing Facilities." AIIE Transactions, 2, 290-297.

[156] Pritsker, A. A. B., and Ghare, P. M. (1970). "Locating New Facilities with Respect to Existing Facilities (Errata and Revisions)." AIIE Transactions, 3, 158-159.

[157] Rado, F. (1988). "The Euclidean Multifacility Location Problem." Operations Research, 36(3), 485-492.

[158] Rao, S. (1973). "On the direct search approach to the rectilinear facilities location problem." AIIE Transactions, 5, 256-264.

[159] Rosen, J. B., and Xue, G. L. (1992). "On the Convergence of Miehles Algorithm for the Euclidean Multifacility Location Problem." Operations Research, 40(1), 188-191.

[160] Rosen, J. B., and Xue, G. L. (1993). "On the Convergence of a Hyperboloid Approximation Procedure for the Perturbed Euclidean Multifacility Location Problem." Operations Research, 41(6), 1164-1171.

[161] Rosenfield, D. B., Engelstein, I., and Feigenbaum, D. (1992). "An Application of Sizing Service Territories." European Journal of Operational Research, 63(2), 164-172.

[162] Rosing, K. E. (1992). "An Optimal Method for Solving the (Generalized) Multi-Weber Problem." European Journal of Operational Research, 58(3), 414-426.

[163] Rutten, W. G. M. M., Van Laarhoven, P. J. M., and Vos, B. (2001). "An extension of the GOMA model for determining the optimal number of depots." Iie Transactions, 33(11), 1031-1036.

[164] Salhi, S., and Gamal, N. D. H. (2003). "A genetic algorithm based approach for the uncapacitated continuous location-allocation problem." Annals of Operations Research, 123(1-4), 203-222.

[165] Scaparra, M. P., and Scutellà, M. G. (2001). "Facilities, locations, customers: building blocks of location models. A survey." Universit a di Pisa Dipartimento di Informatica Technical Report, 1-32.

[166] Sherali, H. D., and Al-Loughani, I. (1998). "Equivalent primal and dual differentiable reformulations of the Euclidean multifacility location problem." IIE Transactions, 30(11), 1065-1074.

[167] Sherali, H. D., and Al-Loughani, I. (1999). "Solving Euclidean distance multifacility location problems using conjugate subgradient and line-search methods." Computational Optimization and Applications, 14(3), 275-291.

[168] Sherali, H. D., Al-Loughani, I., and Subramanian, S. (2002). "Global optimization procedures for the capacitated euclidean and l(p) distance multifacility location-allocation problems." Operations Research, 50(3), 433-448.

[169] Sherali, H. D., Ramachandran, S., and Kim, S. I. (1994). "A Localization and Reformulation Discrete Programming Approach for the Rectilinear Distance Location-Allocation Problem." Discrete Applied Mathematics, 49(1-3), 357-378.

[170] Sherali, H. D., and Shetty, C. M. (1978). "A Primal Simplex Based Solution Procedure for the Rectilinear Distance Multifacility Location Problem " Journal of the Operational Research Society, 29(4), 373-381.

[171] Sherali, H. D., and Tuncbilek, C. H. (1992). "A Squared-Euclidean Distance Location-Allocation Problem." Naval Research Logistics, 39(4), 447-469.

[172] Shetty, C. M., and Sherali, H. D. (1977). "Rectilinear Distance Location-Allocation Problem: A Simplex Based Algorithm." Lecture Notes in Economics and Mathematical Systems, Extremal Methods and Systems Analysis, 174, 442-464.

[173] Smilowitz, K. R., and Daganzo, C. F. (2004). "Cost Modeling and Design Techniques for Integrated Package Distribution Systems."

[174] Taillard, E. D. (2003). "Heuristic methods for large centroid clustering problems." Journal of Heuristics, 9(1), 51-73.

[175] Tuy, H., Khayyal, F. A., and Zhou, F. J. (1995). "A Dc Optimization Method for Single Facility Location-Problems." Journal of Global Optimization, 7(2), 209-227.

[176] Uster, H., and Love, R. F. (2001). "Application of a weighted sum of order p to distance estimation." Iie Transactions, 33(8), 675-684.

[177] Uster, H., and Love, R. F. (2002). "A generalization of the rectangular bounding method for continuous location models." Computers & Mathematics with Applications, 44(1-2), 181-191.

[178] Vardi, Y., and Zhang, C. (2001). "A modified Weiszfeld algorithm for the Fermat-Weber location problem." Mathematical Programming, 90(3), 559-566.

[179] Vergin, R. C., and Rogers, J. D. (1967). "An Algorithm and Computational Procedure for Locating Economic Facilities

[180] " Management Science, 13(6), 240-254.

[181] Verter, V., and Dincer, C. (1995). "Facility Location and Capacity Acquisition: An Integrated Approach." Naval Research Logistics, 42, 1141-1160.

[182] Wang, C. Y., Gao, C. Y., and Shi, Z. J. (1997). "An algorithm for continuous type optimal location problem." Computational Optimization and Applications, 7(2), 239-253.

[183] Ward, J. E., and Wendell, R. E. (1980). "A New Norm for Measuring Distance Which Yields Linear Location-Problems." Operations Research, 28(3), 836-844.

[184] Webster, S., and Gupta, A. (1995). " The general optimal market area model with uncertain and nonstationary demand." Location Science, 3(1), 25-38.

[185] Weiszfeld, E. (1937). "Sur le Point pour Lequel la Somme des Distances de n Points Donn'ees est Minimum." Tohoku Mathematics J. , 43, 355-386.

[186] Wendell, R. E., and Hurter, A. P. (1973). "Location Theory, Dominance, and Convexity." Operations Research, 21(1), 314-320.

[187] Wendell, R. E., and Peterson, E. L. (1984). "A Dual Approach for Obtaining Lower Bounds to the Weber Problem." Journal of Regional Science, 24(2), 219-228.

[188] Wesolowsky, G. O. (1982). "Location problems on a sphere." Regional Science and Urban Economics, 12, 495-508.

[189] Wesolowsky, G. O. (1993). "The Weber problem: History and perspective." Location Science, 1, 5-23.

[190] Wesolowsky, G. O., and Love, R. F. (1971a). "Location of facilities with rectangular distances among point and area destinations." Naval Research Logistics Quarterly, 18, 83-90.

[191] Wesolowsky, G. O., and Love, R. F. (1971b). "The Optimal Location of New Facilities Using Rectangular Distances." Operations Research, 19(1), 124-130.

[192] Wesolowsky, G. O., and Love, R. F. (1972). "A Nonlinear Approximation Method for Solving a Generalized Rectangular Distance Weber Problem " Management Science, 18(11), 656-663.

[193] Xue, G. L. (1995). "On an open problem in spherical facility location." Numerical Algorithms, 9(1), 1-12.

[194] Xue, G. L., Rosen, J. B., and Pardalos, P. M. (1996). "A polynomial time dual algorithm for the Euclidean multifacility location problem." Operations Research Letters, 18(4), 201-204.

[195] Zhao, P. W., and Batta, R. (1999). "Analysis of centroid aggregation for the Euclidean distance p-median problem." European Journal of Operational Research, 113(1), 147-168.